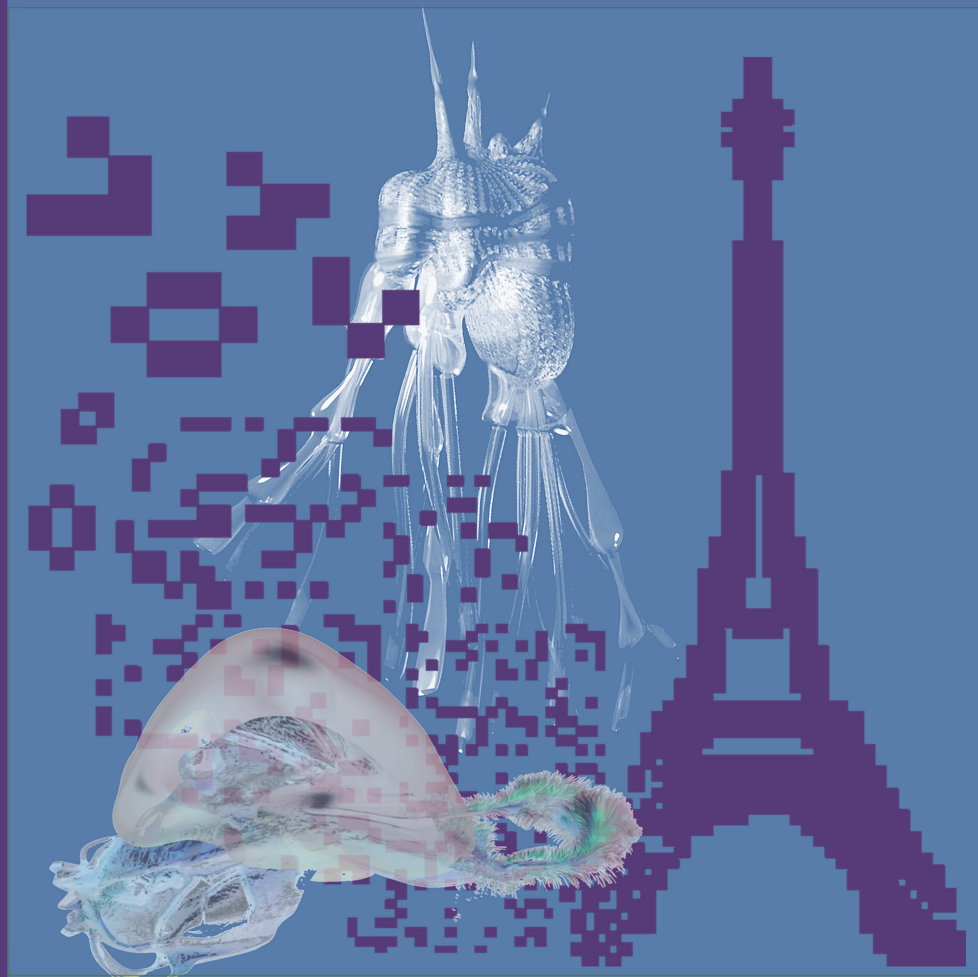


# ADVANCES IN ARTIFICIAL LIFE, ECAL 2011



Proceedings of the Eleventh European Conference  
on the Synthesis and Simulation of Living Systems

edited by

Tom Lenaerts, Mario Giacobini, Hugues Bersini,  
Paul Bourguine, Marco Dorigo, and René Doursat

© 2011 Massachusetts Institute of Technology



This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/3.0/> or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

ISBN 978-0-262-29714-1 (Ebook)

Cover image: Vito Trianni, Manuele Brambilla (CA tower), Louis Bec (creatures) and René Doursat (composition)



*to Francisco Varela (1946-2001),  
Founder of ECAL*

# Table of Contents

<b>Foreword</b>	<b><i>xi</i></b>
-----------------	------------------

<b>Preface</b>	<b><i>xiv</i></b>
----------------	-------------------

## **Invited Contributions**

Jacques Demongeot, Alexandra Henrion-Caude, Athanase Lontos and Emmanuel Promayon. <i>General architecture of a genetic regulatory network. Applications to embryologic control</i> (full article)	1
David Harel. <i>Can we Computerize an Elephant?</i> (abstract)	9
Takashi Ikegami. <i>Do we need a theory in the Era of Massive Data Flow?</i> (abstract)	10
Stuart Kauffman. <i>Answering Descartes: Beyond Turing</i> (full article)	11
James D. Murray. <i>On the Growth of Brain Tumours: enhancing imaging techniques, highlighting limitations of current imaging, quantifying therapy efficacy and estimating patient life expectancy</i> (full article)	23
Jordan Pollack. <i>Prospects for Machine Embryogenesis</i> (abstract)	27
Steen Rasmussen, Pierre-Alain Monnard, Martin Hanczyc, Anders Albertsen, James Boncella, Eva Bonzli, Filippo Caschera, Mark Dorr, Harold Fellermann, Maik Hadorn, Wendie Jørgensen, Philipp Löffler, Sarah Maurer, Kent Nielsen, Pernille Pedersen, Carsten Svaneborg, Michael Wamberg, Rafal Wieczorek and Hans Ziock. <i>A review of one approach to bottom up assembly of minimal life</i> (abstract)	28
Ricard V. Solé and Javier Macia. <i>Synthetic biocomputation: the possible and the actual</i> (full article)	29
Susan Stepney. <i>ALife: a tension between biology and software engineering</i> (abstract)	37
Eric F. Wieschaus, Adam C. Martin, Bing He, Matthias Kaschube and Oleg Polyakov. <i>The mechanics of shape change in the Drosophila embryo</i> (abstract)	38

## **Refereed Articles**

Davide Agostini, Jole Costanza, Vincenzo Cutello, Luca Zammataro, Natalio Krasnogor, Mario Pavone and Giuseppe Nicosia. <i>Effective calibration of artificial gene regulatory networks</i> (full article)	39
Jordi Arranz, Jason Noble and Eric Silverman. <i>The origins of communication revisited</i> (full article)	47
David Balduzzi. <i>Detecting emergent processes in cellular automata with excess information</i> (full article)	55
David J. Barnes and Dominique Chu. <i>Walking, hopping and jumping: a model of transcription factor dynamics on DNA</i> (full article)	63



<i>Artificial gene regulatory networks and spatial computation: A case study</i> (full article)	
Luisa Damiano, Antoine Hiolle and Lola Cañamero. <i>Grounding synthetic knowledge: An epistemological framework and criteria of relevance for the synthetic exploration of life, affect and social cognition</i> (full article)	200
Alastair Droop and Simon Hickinbotham. <i>Application of small-world mutation topologies to an Artificial Life system</i> (abstract)	208
Matthew D. Egbert and Xabier E. Barandiaran. <i>Quantifying normative behaviour and precariousness in adaptive agency</i> (full article)	210
Mehmet D. Erbas and Alan F. T. Winfield. <i>On the emergence of structure in behaviours evolved through embodied imitation in a group of robots</i> (full article)	218
Alessandro Filisetti, Alex Graudenzi, Roberto Serra, Marco Villani, Davide De Lucrezia and Irene Poli. <i>The role of energy in a stochastic model of the emergence of autocatalytic sets</i> (full article)	226
Jonathan M. Fisher and Jason H. Moore. <i>Distinguishing the effects of epistasis and pleiotropy using a variant of the NK model</i> (full article)	234
Regina Frei. <i>A complex systems approach to education in Switzerland</i> (full article)	242
Tom Froese, Nathaniel Virgo and Takashi Ikegami. <i>Life as a process of open-ended becoming: Analysis of a minimal model</i> (full article)	250
Philip Gerlee and Torbjörn Lundh. <i>Urdar - an artificial ecology platform</i> (abstract)	258
Amr S. Ghoneim, Daryl L. Essam and Hussein A. Abbass. <i>On computations and strategies for real and artificial systems</i> (full article)	260
Chris Gordon-Smith. <i>Non-template molecules designed for open-ended evolution</i> (full article)	268
Laura M. Grabowski, David M. Bryson, Fred C. Dyer, Robert T. Pennock and Charles Ofria. <i>Clever creatures: Case studies of evolved digital organisms</i> (full article)	276
Robin Gras, Abbas Golestani, Meisam Hosseini Sedehi, Marwa Khater, Yasaman Majdabadi Farahani, Morteza Mashayekhi, Sina Md Ibne, Elham Salehi and Ryan Scott. <i>EcoSim: an individual-based platform for studying evolution</i> (abstract)	284
Andrew Guest, Andrew Sapeluk, Alan Winfield and James Bown. <i>Promoting meme diversity and transmission fidelity in artificial proto-cultures</i> (full article)	286
Yukio-Pegio Gunji, Hisashi Murakami, Takayuki Niizato, Andrew Adamatzky, Yuta Nishiyama, Koichiro Enomoto, Masashi Toda, Toru Moriyama, Tetsuya Matsui and Kojiro Iizuka. <i>Embodied swarming based on back propagation through time shows water-crossing, hourglass and logic-gate behaviors</i> (full article)	294
Heiko Hamann, Thomas Schmickl and Karl Crailsheim. <i>Explaining emergent behavior in a swarm system based on an inversion of the fluctuation theorem</i> (full article)	302
Taichi Haruna. <i>Global structure of directed networks emerging from a category theoretical formulation of the idea "Objects as processes, interactions as interfaces"</i> (full article)	310
Inman Harvey. <i>Opening stable doors: complexity and stability in nonlinear systems</i> (full article)	318

Mark Hatcher, Wolfgang Banzhaf and Tina Yu. <i>Bondable cellular automata</i> (full article)	326
Simon Hickinbotham, Susan Stepney, Adam Nellis, Tim Clarke, Ed Clark, Mungo Pay and Peter Young. <i>Embodied genomes and metaprogramming</i> (full article)	334
Imad Hoteit, Nawwaf Kharma and Luc Varin. <i>A gene regulatory network design of a synchronous single-input delay flip-flop</i> (full article)	342
Emmanouil Hourdakis and Panos Trahanias. <i>Computational modeling of online reaching</i> (full article)	348
Tim Hoverd and Susan Stepney. <i>Energy as a driver of diversity in open-ended evolution</i> (full article)	356
Takashi Ikegami, Mizuki Oka and Hirotake Abe. <i>Autonomy of the internet: Complexity of flow dynamics in a packet switching network</i> (full article)	364
Takeshi Ishida. <i>Simulation of cell-like self-replication phenomenon in a two-dimensional hybrid-cellular automata model</i> (full article)	372
Daniel Jones and Tim Blackwell. <i>Social learning and evolution in a structured environment</i> (full article)	380
Yoshiaki Katada. <i>Evolutionary dynamics of GAs in a simple model with dynamical environment and neutrality</i> (full article)	388
Susan Khor. <i>Why aren't protein residue networks smaller worlds?</i> (full article)	397
Anaïs Khuong, Guy Theraulaz, Christian Jost, Andrea Perna and Jacques Gautrais. <i>A computational model of ant nest morphogenesis</i> (full article)	404
Joon Kim, Juyeol Yun, Jewoo Hong, Hyojung Kwon and Junghwa Chun. <i>Ecohydrologic and biogeochemical process networks in forest ecosystems in monsoon east asia: Identification and interpretation</i> (abstract)	412
Carole Knibbe, David P. Parsons and Guillaume Beslon. <i>Parsimonious modeling of scaling laws in genomes and transcriptomes</i> (abstract)	414
Davis Knox and John Rieffel. <i>Scalable co-evolution of soft robot properties and gaits</i> (full article)	416
Artemy Kolchinsky and Luis M. Rocha. <i>Prediction and modularity in dynamical systems</i> (full article)	423
Peter Kreyssig and Peter Dittrich. <i>Reaction flow artificial chemistries</i> (full article)	431
Yutetsu Kuruma, Toshiharu Suzuki, Masasuke Yoshida and Takuya Ueda. <i>Artificial organelle for energy production in artificial cell</i> (abstract)	438
Tüze Kuyucu, Ivan Tanev and Katsunori Shimohara. <i>Genetic transposition inspired incremental genetic programming for efficient coevolution of locomotion and sensing of simulated snake-like robot</i> (full article)	439
Sébastien Le Yaouanq, Pascal Redou, Christophe Le Gal, Jean-François Abgrall and Jacques Tisseau. <i>Multi-Agent systems and heterogeneous scales interactions. Application to pharmacokinetics of Vitamin K antagonists</i> (full article)	447
Shuguang Li, Jianping Yuan and Juan Cristóbal Zagal. <i>Encouraging networks modularity</i>	455

*by seeding motifs* (full article)

Joseph T. Lizier, Siddharth Pritam and Mikhail Prokopenko. <i>Computational capabilities of small-world Boolean networks</i> (abstract)	462
Michael A. Lones, Andy M. Tyrrell, Susan Stepney and Leo S. Caves. <i>Controlling legged robots with coupled artificial biochemical networks</i> (full article)	465
Angelo Loula, Ricardo Gudwin and João Queiroz . <i>Cognitive conditions to the emergence of sign interpretation in artificial creatures</i> (full article)	473
Tadao Maekawa, Osamu Ueno, Norie Kawai, Emi Nishina, Manabu Honda and Tsutomu Oohashi. <i>Evolutionary acquisition of genetic program for death</i> (full article)	481
Pedro Mariano and Luís Correia. <i>Evolution of partner selection</i> (full article)	487
Omer Markovitch and Doron Lancet. <i>Evolutionary attributes of simulated prebiotic metabolic networks</i> (full article)	495
Chris Marriott and Carlos Gershenson. <i>Polyethism in a colony of artificial ants</i> (full article)	498
Georg Martius and J. Michael Herrmann. <i>Tipping the scales: guidance and intrinsically motivated behavior</i> (full article)	506
Eiko Matsuda, Julien Hubert and Takashi Ikegami. <i>A robotic approach to understand the role of vicarious trial-and-error in a T-maze task</i> (full article)	514
Luke McCrohon and Olaf Witkowski. <i>Devil in the details: Analysis of a coevolutionary model of language evolution via relaxation of selection</i> (full article)	522
Barry McMullin and James Decraene. <i>Evolution of self-Maintaining cellular Information processing networks</i> (abstract)	530
Dusan Misevic, Charles Ofria and Richard E. Lenski. <i>Digital sex: Causes and consequences</i> (abstract)	532
Takuma Miyake and Kazuto Tominaga. <i>Modeling cell division of B. Subtilis using dynamic division of reaction spaces in a membrane artificial chemistry</i> (full article)	534
Felicitas Mokom and Ziad Kobti. <i>A cultural evolutionary model for artifact capabilities</i> (full article)	542
Jean-Marc Montanier and Nicolas Bredeche. <i>Surviving the tragedy of commons: Emergence of altruism in a population of evolving autonomous agents</i>	550
Hiroataka Moriguchi and Hod Lipson. <i>Learning symbolic forward models for robotic motion planning and control</i> (full article)	558
Jaimie Murdock and Larry S. Yaeger. <i>Identifying species by genetic clustering</i> (full article)	564
Fintan Nagle and Simon Hickinbotham. <i>Embodied reaction logic in a simulated chemical computer</i> (full article)	573
Keita Nakamura, Ikuo Suzuki, Masahito Yamamoto and Masashi Furukawa. <i>Virtual fluid environment on behavior ability for artificial creature</i> (full article)	581
Nikolaos Nanas. <i>Autonomous learning in an information stream through autopoiesis</i> (full article)	589

Adam Nellis and Susan Stepney. <i>Embodied copying for richer evolution</i> (full article)	597
Sancho Oliveira, Luis Nunes and Anders Lyhne Christensen. <i>An Experiment in mixing evolving and preprogrammed robots</i> (full article)	605
Jorge M. Pacheco, Francisco C. Santos, Max O. Souza and Brian Skyrms. <i>Evolutionary dynamics of collective action</i> (abstract)	613
Qinxin Pan, Christian Darabos, Anna L. Tyler, Jason H. Moore and Joshua L. Payne. <i>The influence of whole genome duplication and subsequent diversification on environmental robustness and evolutionary innovation in gene regulatory networks</i> (full article)	614
David P. Parsons, Carole Knibbe and Guillaume Beslon. <i>Homologous and nonhomologous rearrangements: Interactions and effects on evolvability</i> (full article)	622
Jonathan Pascalie, Valérie Lobjois, Hervé Luga, Bernard Ducommun and Yves Duthen. <i>A checkpoint-orientated model to simulate unconstrained proliferation of cells</i> (full article)	630
Joshua L. Payne and Jason Moore. <i>Robustness, evolvability, and accessibility in the signal-integration space of gene regulatory circuits</i> (full article)	638
José N. Pereira, Porfírio Silva, Pedro U. Lima and Alcherio Martinoli. <i>Formalizing institutions as executable petri nets for distributed robotic systems</i> (full article)	646
Enea Pestelacci, Marco Tomassini and Alberto Antonioni. <i>Coordination games on small-worlds: Artificial agents vs. experiments</i> (full article)	654
Raphaël Plasson, Kevin Montagne, Adrien Padirac, Teruo Fujii and Yannick Rondelez. <i>A DNA toolbox for engineering in vitro life-like behaviors</i> (abstract)	662
Arles Rodríguez Portela and Jonatan Gómez Perdomo. <i>Programs self-healing over a termites simulator based on language games and evolutionary computing</i> (full article)	664
Simon T. Powers, Christopher Heys and Richard A. Watson. <i>How to measure group selection in real-world populations</i> (full article)	672
David Roche, Debora Gil and Jesus Giraldo. <i>Using statistical inference for designing termination conditions ensuring convergence of Evolutionary Algorithms</i> (full article)	680
Christoph Salge and Daniel Polani. <i>Local information maximisation creates emergent flocking behavior</i> (full article)	688
Mohammad Samie, Gabriel Dragffy, Tony Pipe and Paul Bremner. <i>An innovative bio-inspired fault tolerant unitronics architecture</i> (full article)	696
Martí Sánchez-Fibla, Armin Duff, Ulysses Bernardet and Paul F.M.J. Verschure. <i>A biomimetic robot controller based on minimizing the unpredictability of the environment: allostatic control revised</i> (full article)	704
Pedro Santana, Ricardo Mendonça, Luís Correia and José Barata. <i>Swarms for robot vision: The case of adaptive visual trail detection and tracking</i> (full article)	712
Francisco C. Santos, Jorge M. Pacheco and Brian Skyrms. <i>Co-evolution of pre-play signaling and cooperation</i> (abstract)	720
Yuki Sato, Hiroyuki Iizuka and Takashi Ikegami. <i>An experimental and computational approach to the dynamic body boundary problem</i> (full article)	721

Hiroki Sayama and Chun Wong. <i>Quantifying evolutionary dynamics of swarm chemistry</i> (full article)	729
Malte Schilling. <i>Learning by seeing -- associative learning of visual features through mental simulation of observed action</i> (full article)	731
Lisa Schramm and Bernhard Sendhoff. <i>An animat's cell doctrine</i> (full article)	739
Eric Silverman, Jakub Bijak and Jason Noble. <i>Feeding the beast: Can computational demographic models free us from the tyranny of data?</i> (full article)	747
Jean Sirmai. <i>A schematic representation of autopoiesis using a new kind of discrete spatial automaton</i> (full article)	755
Jorge Soto-Andrade, Sebastián Jaramillo, Claudio Gutiérrez and Juan-Carlos Letelier. <i>Ouroboros avatars: A mathematical exploration of self- reference and metabolic closure</i> (full article)	763
Michael Spranger. <i>Recruitment, selection and alignment of spatial language strategies</i> (full article)	771
Pasquale Stano, Paolo Carrara, Tereza Pereira de Souza and Pier Luigi Luisi. <i>An update on the minimal cell project: From the physics of solute encapsulation to the experimental modeling of cell communities</i> (abstract)	779
Susan Stepney and Tim Hoverd. <i>Reflecting on open-ended evolution</i> (full article)	781
Alessandro Stranieri, Eliseo Ferrante, Ali E. Turgut, Vito Trianni, Carlo Pinciroli, Mauro Birattari and Marco Dorigo. <i>Self-organized flocking with an heterogeneous mobile robot swarm</i> (full article)	789
James Thorniley. <i>An improved transfer entropy method for establishing causal effects in synchronizing oscillators</i> (full article)	797
Nicholas Tomko, Inman Harvey, Andrew Philippides and Nathaniel Virgo. <i>Many hands make light work: Group evolution and the emergent division of labour</i> (full article)	805
Ryoko Uno, Keisuke Suzuki and Takashi Ikegami. <i>An interactive wall game as an evolution of proto language</i> (full article)	813
Tomas Veloz, Bryan Reynaert, David Rojas-Camaggi and Peter Dittrich. <i>A decomposition theorem in chemical organizations</i> (full article)	820
Nathaniel Virgo, Chrisantha Fernando, Bill Bigge and Phil Husbands. <i>The elongation catastrophe in physical self-replicators</i> (full article)	828
Nikolaos Vlassopoulos and Nazim Fatès. <i>Clustering behavior of a bio-inspired decentralized aggregation scheme</i> (abstract)	836
X. Rosalind Wang, Jennifer M. Miller, Joseph T. Lizier, Mikhail Prokopenko and Louis F. Rossi. <i>Measuring information storage and transfer in swarms</i> (full article)	838
Angela Watkins, Jason Noble and C. Patrick Doncaster. <i>An agent-based model of jaguar movement through conservation corridors</i> (full article)	846
Richard A. Watson, Rob Mills and Christopher L. Buckley. <i>Transformations and multi-scale optimisation in biological adaptive networks</i> (abstract)	854
Hywel T.P. Williams, Richard A. Boyle and Tim M. Lenton. <i>Spatial structure creates</i>	856



<i>community-level selection for nutrient recycling</i> (abstract)	
Lance R. Williams. <i>Artificial cells as reified quines</i> (full article)	858
Peter R. Wills. <i>Life requires genetic representation and vice versa – Consequences for ALife</i> (full article)	866
Shelly Xiaonan Wu and Wolfgang Banzhaf. <i>Evolutionary transition through a new multilevel selection model</i> (full article)	874
Keisuke Yoneda, Ikuo Suzuki, Masahito Yamamoto and Masashi Furukawa. <i>Behavioral acquisition of complicated locomotion for artificial elastic robot using decentralized behavior composed</i> (full article)	882
Jason Yosinski, Jeff Clune, Diana Hidalgo, Sarah Nguyen, Juan Cristobal Zagal and Hod Lipson. <i>Evolving robot gaits in hardware: the HyperNEAT generative encoding vs. parameter optimization</i> (full article)	890
Seung-Eun Yu and DaeEun Kim. <i>Landmark vector model with quantized distance for homing navigation</i> (abstract)	898
Elisabeth Zu Erbach-Schoenberg, Connor McCabe and Seth Bullock. <i>On the interaction of adaptive timescales on networks</i> (full article)	900
<b>Author list</b>	<b>908</b>

# Foreword

## ECAL'11: Back to the Origins of Alife

There is a long tradition of software simulations in theoretical biology to complement pure analytical mathematics, which are often limited in their ability to reproduce and understand self-organisation phenomena resulting from nonlinear and spatially grounded interactions of a huge number of various and evolving biological objects. Researchers in Artificial Life (Alife) bet that they can assist biologists in this domain, transcending their daily modelling and measuring practice by using software simulation in the first instance, and robotics, too, in order to abstract and elucidate the fundamental mechanisms common to living organisms. They hope to do so by discovering the most informative level of abstraction and resolutely neglecting a lot of physical and quantitative information deemed not indispensable. The computer is apparently the best microscope to achieve this. They want to focus on the rule-based mechanisms that make life possible, supposedly neutral with respect to their underlying material embodiment, and replicate them in a non-biochemical substrate. The hypothesis is that minimal life begins at the intersection of a series of processes that need to be isolated, differentiated and duplicated as such in computers, and that only software development and execution make possible to understand the way these processes are intimately interconnected in order for life to emerge at the crossroad.

The rejection of an authoritative definition of “life” is often compensated for by a list of functional properties that never finds unanimity amongst its authors. Some demand more properties, others require fewer of those properties that are often indicated in vague terms such as “self-maintenance”, “self-organisation”, “metabolism”, “autonomy”, “self-replication”, or “open-ended evolution”. A first determining role of Alife consists in writing and implementing software versions of these properties and the way they actually interact. The goal is to disambiguate them and make them algorithmically precise enough so that, in the end, the only remaining cause of disagreement on the definition of life would reside in the length or the composition of this list but not in its items.

Biologists obviously remain the most important partners; but what may they expect from this Alife business? What can they expect from these new “Merlin hackers”, whose ambitions seem, above all, disproportionately naïve. Computer platforms are useful and necessary in several ways. First of all, they open the door to a new style of teaching and advocating major biological ideas: in other words, computer software as a pedagogical tool. For example, Richard Dawkins is the best advocate of Darwinian ideas when running a computer simulation in which sophisticated creatures known as “biomorphs” evolve on a computer screen by means of a genetic algorithm. These same platforms and simulations can, insofar as they are sufficiently flexible, quantifiable and universal, be used more accurately by biologists, who will find in them a simplified way of simulating and validating a given biological system under study. Cellular automata, Boolean networks, genetic algorithms and algorithmic chemistry are excellent examples of softwares to download, parameterise and use to reproduce the required natural phenomenon. Their predictive

power varies from very qualitative (where results apparently reproduce general trends of the real world) to very quantitative (where numbers produced by the computer may be precise enough to be compared with those measured in the real world).

Although being at first very qualitative, precise and clear coding is already the guarantee of an advanced understanding accepted by all. Algorithmic writing is an essential stage in formalising the elements of the model and making them objective. The linguistic and qualitative style of many biological papers could benefit in clarity by attempting a software instantiation of their contents. The more the model allows to integrate what we know about the reality being reproduced, the detailed structure of the objects and the relationships between them, the more the predictions will move from qualitative to precise, and the more easily the model will be validated according to Karl Popper's "falsification" process of good scientific practice.

Ideally, through systematic software experiments, these platforms can lead to the discovery of new natural laws, whose impact will be all the greater as the simulated abstractions will be present in many biological realms. In the 1950's, when Alan Turing discovered that a simple diffusion phenomenon, propagating itself at different speeds, depending on whether it is subject to a negative or positive influence, produces zebra or alternating motifs, it had a considerable effect on a whole section of biology studying the genesis of forms. This was Alife at its best. The same happened with John von Neumann's self-replicating automata. Because of these seminal works, Turing and von Neumann remain the two spiritual fathers of the field. When scientists discovered that the number of attractors in a Boolean network (Kauffman) or a neural network (Hopfield) exhibited a given dependency on the number of units in these networks, these results could equally well apply to the number of cell types expressed as dynamic attractors in a genetic network or the quantity of information capable of being memorised in a neural network. Entire chapters of biology dedicated to networks (neural, genetic, protein, immune, hormonal) had to be re-written in the light of these discoveries. When other scientists recently observed non-uniform connectivity in many networks, whether social, technological or biological, showing a small number of hubs with a large number of connections and a greater number of nodes with far fewer, and when, in addition, they explained the way in which these networks are built in time by preferential attachment (Barabási), again biology was clearly affected.

Alife is of course at its best when it reveals new biological facts, destabilizing biologists' presuppositions or generating new knowledge, rather than simply illustrating or refining the old. Roughly said, we could construe Alife as being to theoretical biology what mathematics is to physics, that is, a more neutral scientific endeavour to provide open-minded biologists with new tools and new formal terms to describe and conceptualize the objects of their study. At the moment, the fact that this discipline is still young and shows relative immaturity in comparison with mainstream biology might explain why some observers remain skeptical in front of the current discrepancy between promises and reality. In our opinion, however, they tend to underestimate the importance of the results already obtained, as they are too riveted to their microscope. They should show less reluctance, indifference or even arrogance – and more

curiosity and conviviality – towards these new computer explorers who have set out on the conquest of life, just like them but in front of their computer screens.

This is how we saw Alife when Francisco Varela, Paul Bourguine and myself decided to organize the first European Conference on Artificial Life (ECAL) in Paris twenty years ago. We were very impressed by the Sante Fe workshops, which Chris Langton had started, and decided to initiate a European counterpart. We were aware of this very long tradition of theoretical biology in Europe, nevertheless a tradition still largely unaffected by the new opportunities offered by software simulations. We also realized that an opportunity existed to expand Alife “toward a practise of autonomous systems” with their “embodied cognition”, including not only all forms of life but also autonomous robots and collective intelligence. We emphasised the importance of developing artificial life toward new trends in theoretical biology, based on such practise of autonomous systems and not only on purely literary descriptions or purely mathematical formulations.

This opportunity exists more than ever for the future and we wanted to provoke discussions at ECAL about all major forms of autonomous systems, characterized by self-organized architectures, morphogenesis and adaptation, from minimal forms of life to the ecosphere, from minimal forms of cognition to human social intelligence, mediated through internet and the web. Besides, we did not want artificial life to become a sub-branch of engineering only weakly inspired by biology. In fact, other conferences already existed for that. ECAL ought to be different and unique, genuinely centred on theoretical biology and the physics of complex autonomous systems.

Today, although we are proud of this series of very successful and exciting ECAL conferences, we feel that the domain of Alife should look back to these origins and take even more inspiration from the new high-throughput developments at the intersection between computer science and theoretical biology. Closing a loop, this year’s ECAL will mark the 20th anniversary of the first ECAL and will be framed as a tribute to the late Francisco Varela. It was summer 1990, the three of us, with Paul Bourguine, were sitting in a café in Paris, drinking an excellent wine, when Francisco proposed to make our own version of an Alife conference. Thanks Francisco, we miss you.

Welcome to ECAL 2011!

Hugues Bersini

*Brussels  
August 2011*

## Preface

ECAL, the European Conference on Artificial Life, is a biennial event that alternates with the US-based Alife conference series. In the early 1990's, the first two ECAL conferences in Paris and Brussels were mainly centered on theoretical biology and the physics of complex systems. After 20 years and 10 editions of this event, we felt that the domain should look back on these origins and our wish was to refocus the ECAL conference on complex biological systems.

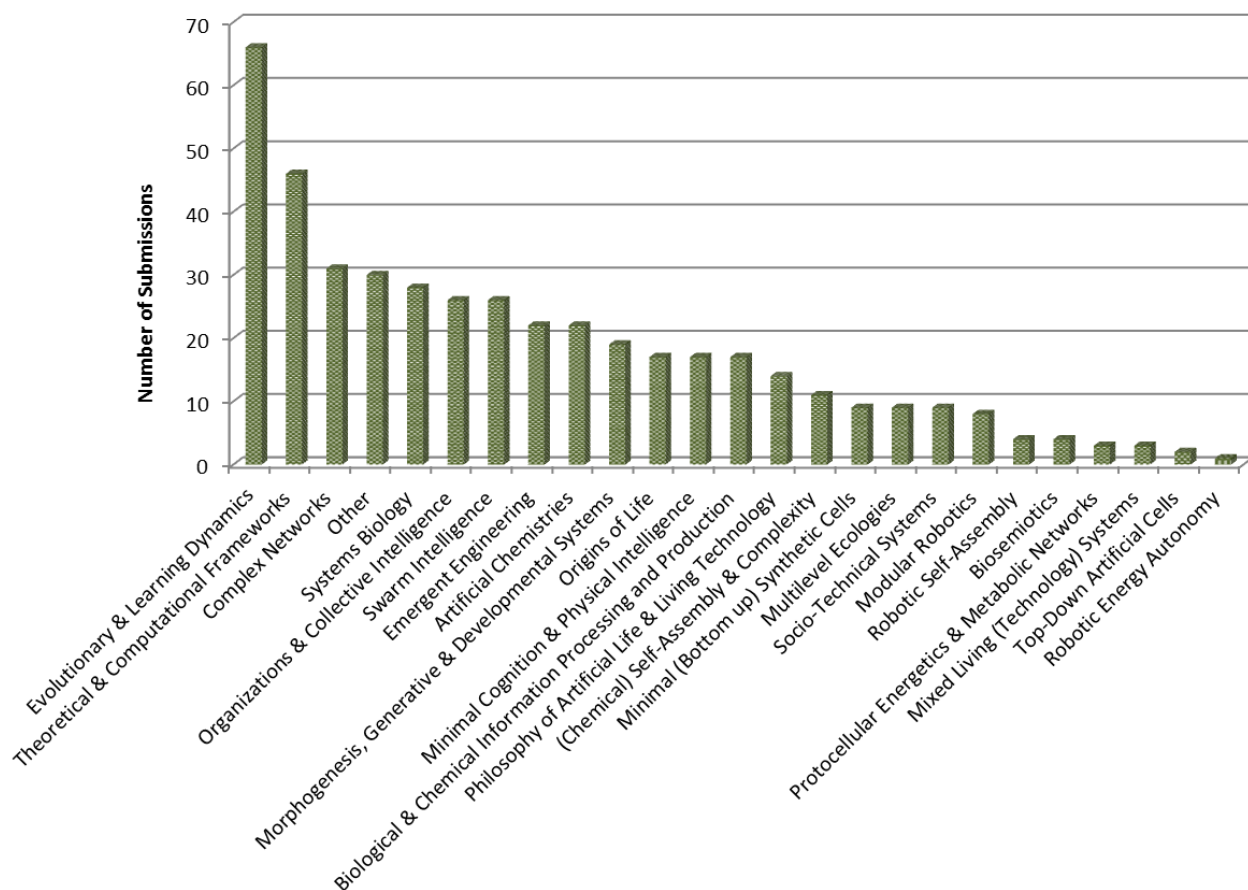
Over the past two decades, biological knowledge has grown at an unprecedented rate, giving rise to new disciplines such as systems biology, testimony of the striking progress of modeling and quantitative methods across the field. During the same period, highly speculative ideas have matured, and entire conferences and journals are now devoted to them. Synthesizing artificial cells, simulating large-scale biological networks, storing and making intelligent use of an exponentially growing amount of data (e.g., microarrays), exploiting biological substrates for computation and control, and deploying bio-inspired engineering are all cutting-edge topics today.

ECAL'11 leveraged this remarkable development of biological modeling and extended the topics of Artificial Life to the fundamental properties of living organisms: their multiscale pattern-forming morphodynamics, their autopoiesis, robustness, capacity to self-repair, cognitive capacities, and co-adaptation at all levels, including ecological ones. Bringing together a large interdisciplinary community of biologists, computer scientists, physicists, and mathematicians, the conference gave them a moment to reflect on how traditional boundaries between disciplines have become blurred, and to revisit in depth what constitutes "life".

In order to make the event attractive to researchers from a wide range of disciplines, we decided to open the possibility to submit 2-page abstracts discussing work previously published by the authors in a journal. In addition to 148 full-length (8-page) articles reporting on new, unpublished work, we received 29 overview abstracts, for a total of 177 submissions.

Although intrinsically interdisciplinary, these submissions referred in particular to the main conference topics, as described by the histogram below. All submissions were subject to peer review. The work of our excellent Program Committee (see list of members below) allowed us to select 128 papers, subdivided into 72 oral presentations (for a 40.7% acceptance rate) and 56 posters (31.6%), with no distinction being made between the two submission options, full paper or abstract. Two accepted papers were later withdrawn by their authors, reducing the total number to 126 (72 + 54).

All papers were presented during the four days of the plenary conference, which was held at the Cité Internationale Universitaire de Paris, France from August 9 to 12, 2011. Oral and poster sessions alternated with six world-class keynote speakers, whose invited contributions (abstracts or full papers) are also published in the front section of these proceedings: Jacques Demongeot, David Harel, James D. Murray, Jordan Pollack, Ricard Solé, and Eric Wieschaus (see their biosketches below). We thank them for taking the time and effort to participate in the conference.



## Satellite Workshops

In addition to the plenary conference, we were also pleased to give Alife researchers the opportunity to organize satellite workshops and tutorials in two “bookend” sessions, on the first day (August 8) and last afternoon (August 12). These special sessions were dedicated to the same [general topics](#) as the main conference, while allowing for more focused interactions among participants. They were independently managed by their organizers and could comprise any combination of peer-reviewed papers, posters, invited talks, panel discussions, etc. Workshop contributions were not included in these proceedings. We received 15 proposals, of which 14 effectively took place, testimony of the liveliness of the field:

- *AAALE: Alife Approaches to Artificial Language Evolution*  
Luc Steels and Tony Belpaeme
- *ACCS: Artificial Chemical Computing Systems*  
Hideaki Suzuki and Hiroki Sayama
- *BioChemIT: 1st COBRA Workshop on Biological and Chemical Information Technologies*  
Peter Dittrich, Zarka Khan and Martyn Amos

- *CoSMoS: 4th Workshop on Complex Systems Modelling and Simulation*  
Paul Andrews, Susan Stepney, Peter Welch and Carl Ritson
- *CS-Sports: Complex Systems in Sports*  
Juan Julián Merelo Guervós, Antonio Mora García and Carlos Cotta Porras
- *DDLab: Exploring Discrete Dynamics: From Cellular Automata to Random Networks*  
Andy Wuensche, Andy Adamatzky and Genaro Juárez Martínez
- *iBioMath: First International Workshop on Integral Biomathics*  
Plamen Simeonov, Andrée Ehresmann and Leslie Smith
- *INCUP: Information Coding in Unconventional Computing Substrates*  
Jerzy Gorecki and Andy Adamatzky
- *MASmms: Workshop on Multi-Agent Systems in Biology at meso or macroscopic scales*  
Pascal Ballet, Marie Beurton-Aimar, Guillaume Hutzler and Bertrand Laforge
- *MEW: 3rd Morphogenetic Engineering Workshop*  
René Doursat and Hiroki Sayama
- *RUTSAC: Research Using The Stringmol Artificial Chemistry*  
Simon Hickinbotham, Ed Clark and Adam Nellis
- *SIM-A: System Immunology Models of Autopoiesis*  
Uri Hershberg and Sol Efroni
- *SynBioCCC: Workshop on the Design, Simulation, Construction and Testing of Synthetic Gene Regulatory Networks for Computation, Control and Communications*  
Nawwaf Kharma and Taras Kowaliw
- *WAAT: Workshop on Artificial Autonomy: 20 years of practice of autonomous systems*  
Tom Froese, Matthew Egbert and Xabier Barandiaran

## Keynote Speakers

- **Jacques Demongeot** is presently director of the TIMC-IMAG Laboratory, “Techniques of Medical Engineering & Complexity” (CNRS 5525) and is also head of the Institute of Bioengineering (IFRT 130 IpV) at the University Joseph Fourier, Grenoble, France. He has an MD and a PhD in mathematics and has been appointed Chairman of Biomathematics at the Institut Universitaire de France in 1994. Jacques Demongeot is also in charge of the Department of Medical Information at the University Hospital of Grenoble (CHUG) and is the founder of the doctoral school of bioengineering “Health, Cognition & Environment”. He is currently creating a new laboratory AGIM, in Archamps near Geneva, devoted to studies of development and ageing.
- **David Harel** is a professor of computer science at the Weizmann Institute of Science in Israel. Harel is best known for his work on dynamic logic, computability and software engineering. In the 1980s he invented the graphical language of Statecharts, which has been adopted as part of the UML standard. He has also published expository accounts of computer science, such as his award winning 1987 book “Algorithmics: The Spirit of Computing” and has made appearances on Israeli

radio and television. He currently works on many diverse topics, including visual languages, graph layout, systems biology and the communication of odours. He is now working on a computer model of a nematode, ‘*Caenorhabditis elegans*’, which was the first multicellular organism to have its genome completely sequenced. The eventual completeness of such a model depends on his updated version of the test developed by Alan Turing to identify whether computers could reason well enough that a human communicating with them could not tell whether a human or a machine was at the other end of the communication.

- **James D. Murray**, FRS, Foreign Member of the French Academy, is Professor Emeritus of Mathematical Biology at the University of Oxford, Professor Emeritus of Applied Mathematics at the University of Washington, and Senior Scholar at Princeton University. His research is characterized by its great scope and depth: an early example is his fundamental contributions to understanding the biomechanics of the human body when launched from an aircraft in an ejection seat. He has made contributions to many other areas, ranging from understanding and preventing severe scarring, to fingerprint formation, sex determination, modeling of animal coat patterns, territory formation in wolf-deer interacting populations, growth and control of brain tumors, quantifying patient treatments prior to use, and modeling marital interaction and divorce prediction with 94% accuracy in a 12-year longitudinal study. He is best known for his authoritative and extensive work entitled *Mathematical Biology*, whose 3rd edition in two volumes came out in 2004.
- **Jordan Pollack** is professor of computer science and complex systems professor at Brandeis University, where he is also chairman of the computer science department and director of the Dynamical and Evolutionary Machine Organization (DEMO) lab. The laboratory’s work on AI, Artificial Life, Neural Networks, Evolution, Dynamical Systems, Games, Self-designed Robotics, Machine Learning, and Educational Technology has been reported on by the New York Times, Time, Science, NPR, Slashdot.org and many other media sources worldwide.
- **Ricard Solé** heads the Complex Systems Lab at Universitat Pompeu Fabra, Barcelona, and is an External Professor at the Santa Fe Institute. One of his main research interests is understanding the possible presence of universal patterns of organization in complex systems, from prebiotic replicators to evolved artificial objects. Key questions are how robust structures develop, how information is incorporated into these structures and how computation emerges. He is also interested in how to determine what are the contributions of selection, chance and self-organization to the evolution of complexity. One of his main goals is searching for the principles of organization responsible for the emergence of fundamental components of complexity, including the origins of self-reproduction, development, life cycles, computational processes and multicellularity. His work has been featured in newspapers as well as several popular and technical books.
- **Eric Wieschaus** is the Squibb Professor in Molecular Biology at Princeton. His research work has focused on embryogenesis in the fruit fly *Drosophila melanogaster*, specifically in the patterning that occurs in the early *Drosophila* embryo. Most of the gene products used by the embryo at these stages are already present in the unfertilized egg and were produced by maternal transcription during oogenesis. A small number of gene products, however, are supplied by transcription in the embryo itself. He has focused on these “zygotically” active genes because he believes the temporal and spatial pattern of their transcription may provide the triggers controlling the normal sequence of embryonic development. In 1995, he was awarded the Nobel Prize in Physiology or Medicine with Edward B. Lewis and Christiane Nüsslein-Volhard as co-recipients, for their work revealing the genetic control of embryonic development.



## Alife Pioneers Panel Discussion

In addition to an exceptional selection of keynote speakers, an exciting panel discussion involving several internationally renowned pioneers of Artificial Life took place at the end of the second day of the plenary conference. Mark Bedau, Takashi Ikegami, Stuart Kauffman, Norman Packard, Steen Rasmussen, Luc Steels, and Susan Stepney all talked about the most impressive achievements of Alife in the past, since inception of the field, and pointed to what they thought were the most promising research directions for the future. Some of them also provided an invited contribution (abstract or full paper), which can be found in the same section as the keynotes' contributions.

- **Mark Bedau** (Reed College, Portland – European School of Molecular Medicine, Milan – Initiative for Science, Society, and Policy, Denmark) pioneered the field of quantifying and comparing the evolutionary activity in artificial and natural systems, and is an international leader in the evolutionary design of complex biochemical systems using statistical models and prediction algorithms. Because he combines training in analytical philosophy with over a decade of experience in artificial life, he is recognized as a uniquely qualified expert in the philosophical foundations of complex adaptive systems. Mark Bedau is Editor-in-Chief of the international journal *Artificial Life* (published by MIT Press), he co-organized five international conference on artificial life, co-founded a start-up company, ProtoLife SRL, and co-founded the European Center for Living Technology, a research institute in Venice, Italy, that investigates theoretical and practical issues associated with living systems.
- **Takashi Ikegami** is a professor at the Department of General Systems Sciences of the Graduate School of Arts and Sciences, University of Tokyo, where he specializes in complex systems and artificial life. Takashi takes a computational/philosophical approach to designing artificial life, exploring issues at the margins of his discipline. He is also an arts collaborator with Keichiro Shibuya (ATAK) on making three-dimensional sound installations. Keywords: chemical computing, smart chemical agents, chemotaxis, living technology, artificial life, first cell.
- **Stuart Kauffman** (University of Vermont, Burlington) is an American theoretical biologist and complex systems researcher concerned with the origin of life on Earth. He is best known for arguing that the complexity of biological systems and organisms might result as much from self-organization and far-from-equilibrium dynamics as from Darwinian natural selection, as well as for applying models of Boolean networks to genetic circuits. Stuart Kauffman rose to prominence through his association with the Santa Fe Institute, where he was faculty in residence (1986-1997), and his work on models in various areas of biology. These included autocatalytic sets in origin of life research, gene regulatory networks in developmental biology, and fitness landscapes in evolutionary biology. Stuart Kauffman held a joint appointment at the University of Calgary in Biological Sciences and Physics and Astronomy (2005-2009), then joined in 2010 the University of Vermont where he will continue his work with UVM's Complex Systems Center.
- **Barry McMullin's** primary research activity at the Rince Research Institute, Dublin City University (DCU), is in the domain of Artificial Life. He serves on the organizing committees of both ECAL and Alife conferences, and as a member of the Editorial board of the *Artificial Life* journal. He has a secondary research interest in the area of Web Accessibility, engineering web sites and services to best meet the requirements of all users, specifically including those with disabilities. Between 1999 and 2004, Barry McMullin was the first DCU Dean of Teaching and Learning. In this role he was responsible for the development of a wide series of initiatives to significantly enhance the quality of the student learning experience at DCU. Barry McMullin was

appointed to the rank of Associate professor at DCU in September 2010, and became Director of RINCE, a national research institute specializing in Engineering technology innovation, in February 2010.

- **Norman Packard** (European Center for Living Technology, Venice) has worked in the areas of chaos, learning algorithms, predictive modeling of complex time series, statistical analysis of evolution, artificial life, and complex adaptive systems. He was co-founder of Prediction Company in 1991 and served as its CEO (1997-2003) and chairman until 2005. Norman Packard is currently working in a new scientific and business direction based on development of evolutionary chemistry in programmable microfluidic technology. Long-range applications of this technology include the fabrication artificial cells from non-living material, and their programming for useful functionality. In 2004, Norman Packard was co-founder of ProtoLife S.r.l. (Venice, Italy), which applies machine learning techniques to the design of experiments (DoE) for high throughput experiments in biotechnology. As part of the PACE project (Programmable Artificial Cell Evolution, 2004-2008), he also participated in the founding of ECLT, the European Center for Living Technology.
- **Steen Rasmussen** is currently the Head of the Center for Fundamental Living Technology (FLinT), a Research Director at the Department for Physics and Chemistry at University of Southern Denmark, Odense, and External Research Professor at the Santa Fe Institute. He has pioneered approaches, methods, and applications for self-organizing processes in natural and artificial systems: abstract self-programmable matter, molecular dynamics (MD) lattice gas simulations for molecular self-assembly, rational and evolutionary protocell design, disaster mitigation and decision support systems based on collective intelligence, as well as novel simulations for large-scale sociotechnical systems. Steen Rasmussen was heading the Protocell Assembly (LDRD-DR) project and the Astrobiology program (origins of life) at Los Alamos, developing experimental and computational protocells and cell-like entities. He also co-directed the European PACE project (Programmable Artificial Cell Evolution) project.
- **Luc Steels** is professor of Computer Science (at the moment part-time) at the Free University of Brussels (VUB), founder and director (since 1983) of the VUB Artificial Intelligence Laboratory and co-founder and chairman (1990-1995) of the VUB Computer Science Department. He has also been the director of Sony CSL in Paris since its creation in 1996. His scientific research interests cover the whole field of artificial intelligence, including natural language, vision, robot behavior, learning, cognitive architecture, and knowledge representation. At the moment his focus is on dialogs for humanoid robots and fundamental research into the origins of language and meaning. Current work focuses on developing the foundations of semiotic dynamics and on fluid construction grammars.
- **Susan Stepney** leads the Non-Standard Computation research group, and is one of the instigators of the new interdisciplinary York Centre for Complex Systems Analysis, University of York. Originally a theoretical astrophysicist, she has spent the bulk of her professional career in industrial R&D (GEC-Marconi and Logica), mostly in mathematical and computational modelling, researching aspects of novel computation. She is a moderator of the UKCRC Grand Challenge 7 in Non-Classical Computation and is helping to build a conceptual meta-framework for bio-inspired computation. Current research interests also include theories of emergence and self-organising systems, and nature-inspired computational metaphors. She is the PI of the Complex Systems Modelling and Simulation project and was PI of the EIVIS novel computation cluster, rated “outstanding”. She also teaches complex biosystems simulation and is responsible for designing the new Masters course in Natural Computation at York.

## Alife Art Exhibit and Performance

Inspiration, imagination and aesthetics are an integral part of science, and they are of particular importance in the Alife community, which fuels some of the most creative and provocative research at the edge (of chaos) between biology and technology. Accordingly, ECAL'11 welcomed a prominent visual artist, Louis Bec, and a distinguished musician, François Pachet, who showcased their exciting work in the exhibit rooms and main auditorium.

- **Louis Bec**, born in Algeria and living in France, is a biologist and zoosystematician who extends his scientific field with a fabulatory epistemology based on Artificial Life and Technozoosemiotics. In 1972, Bec founded the Institut Scientifique de Recherche Paranaturaliste, where he studies the incapability of living beings to understand their own existence. Bec is both artist and scientist in the field of artificial life and 3D technologies. He is as much a biologist, artist, curator and educator, and has been a ministry officer for new technologies in arts. Bec is Director of CYPRES (Centre Interculturel de Pratiques et Echanges Transdisciplinaires) in Marseille. He has presented his ideas in many exhibitions, such as Alife II (invited by Chris Langton) and *From Animals to Animats*, and articles.
  - ***Upokrinomenes: a fabulated epistemology*** Zoosystemician Louis Bec forces us to question the validity of each claim by reformulating and staging scientific discourse. His reasoning possesses all the marks of scientific assertiveness, combining scientific jargon with scholarly neologisms. Questioning life and our inability to understand it through traditional investigative methods, he founded the field of *Upokrinomenology*. It is a theory of life using models based on computer science, robotics, video and other interactive devices, where irony holds a significant place. By putting scientific discourse into perspective, he challenges us to investigate, unravel and interpret the propositions that he makes. In this research, scientific discourse becomes poetic and Louis Bec becomes a storyteller. Founder of the Scientific Institute of Paranaturalistic Research, he invites us to discover a life we did not know existed, one that looms at the border between shapes, language and behavior [after C. Beaugrand & A. Charre, *Reinventing the museum*]. (Art exhibit at ECAL'11 designed and installed with François Mourre, Patrice Bersani and Delphine Fabbri-Lawson.)
- **François Pachet** is a Civil Engineer (Ecole des Ponts and Chaussées) and was an Assistant Professor in Artificial Intelligence and Computer Science, University of Paris 6, until 1997. He then set up the music research team at the Sony Computer Science Laboratories, Paris, and developed the vision that metadata can greatly enhance the musical experience in all its dimensions, from listening to performance. His team conducts research in interactive music listening and performance and musical metadata and developed several innovative technologies and award winning systems (MusicSpace, constraint-based spatialization, PathBuilder, intelligent music scheduling using metadata, The Continuator for Interactive Music Improvization). He is the author of over 80 scientific publications in the fields of musical metadata and interactive instruments.
  - ***The Continuator Project: playing with virtual musicians*** François Pachet (guitar player) and Jeff Suzda (professional sax player) performed a short Jazz concert with their band “Quintet of Two”. They comprised the two human musicians in the group, performing alongside three “software” musicians. The goal of this project was to play “standard” jazz using virtual instruments intimately controlled by the human players. The technologies employed, developed at Sony CSL, involve Markov chains, constraint programming, signal processing, and a great degree of musical tuning. Performance was still exploratory, but the goal is to enhance musical expressivity through controllable machines.

## Program Committee

An event like ECAL'11 would not have been possible without the following dedicated members of the Program Committee and additional reviewers. Our gratitude goes to all of them.

Hussein Abbass	Joachim De Beule	Pedro U. Lima	Matthias Scheutz
Andy Adamatzky	Bart De Boer	Daniel Lobo	Thomas Schmickl
Chris Adami	Ralf Der	Fernando Lobo	Marc Schoenauer
Andreas Albrecht	Gianni Di Caro	Robert Lowe	Luis Seabra Lopes
Fernando Almeida e Costa	Cecilia Di Chio	Penousal Machado	Roberto Serra
Lee Altenberg	Peter Dittrich	Steven Maere	Cosma Shalizi
Paul Andrews	Marco Dorigo	Davide Marocco	Ricard Sole
Takaya Arita	Alan Dorin	John McCaskill	Antoine Spicher
Wolfgang Banzhaf	René Doursat	Chris McEwan	Peter Stadler
Xabier E. Barandiaran	Marc Ebner	Barry McMullin	Susan Stepney
Andrea Baronchelli	Arantza Etxeberria	Jose Mendes	Charles Taylor
Jacob Beal	Nazim Fates	Olivier Michel	Tim Taylor
Mark Bedau	Christoph Flamm	Martin Middendorf	Gianluca Tempesti
Randall Beer	Luca Gambardella	Luís Moniz Pereira	Christof Teuscher
Tony Belpaeme	Carlos Gershenson	Sara Montagna	Jon Timmis
Peter Bentley	Mario Giacobini	Jason Moore	Peter Todd
Hugues Bersini	Roderich Gross	Federico Moran	Marco Tomassini
Luc Berthouze	Thilo Gross	Chrystopher L. Nehaniv	Arne Traulsen
Luigi Bertolotti	Pauline C Haddow	Jason Noble	Vito Trianni
Mauro Birattari	Emma Hart	Stefano Nolfi	Elio Tuci
Joris Bleys	Inman Harvey	Ann Nowe	Gunnar Tufte
Josh Bongard	Paulien Hogeweg	Charles Ofria	Ali Emre Turgut
Paul Bourguin	Phil Husbands	Joshua Payne	Karl Tuyls
Seth Bullock	Fumiya Iida	Andrew Philippides	Jon Umerez
Stefano Cagnoni	Yaochu Jin	Raphaël Plasson	Patricia A. Vargas
Alexandre Campo	Colin Johnson	Daniel Polani	Mirko Viroli
Philippe Capdepuy	Istvan Karsai	Paolo Provero	Paul Vogt
Ciro Cattuto	Jozef Kelemen	Charles Richter	Richard Watson
Anders Christensen	Serge Kernbach	Luis M. Rocha	Alan Winfield
Dominique Chu	Daeun Kim	Miguel Rocha	Rachel Wood
Netta Cohen	Taras Kowaliw	Pierre Rouze	Andrew Wuensche
Luis Correia	Kalevi Kull	Kepa Ruiz Mirazo	Larry Yaeger
Ernesto Costa	Renaud Lambiotte	Erol Sahin	Tom Ziemke
Tamás Czárán	Doron Lancet	Francisco C. Santos	
Christian Darabos	Tom Lenaerts	Hiroki Sayama	

## Additional Reviewers

Philipp Altmann	Nanda Maheshwari	Gabriel Piedrafita	Nikolaos Vlassopoulos
Joshua Auerbach	Pedro Mariano	Flavio Pinheiro	Liyu Wang
Julian Garcia	Hugi Marques	Stuart Rossiter	Nicolae-Radu Zabet
Heather Goldsby	Bjørn Østman	Pedro Santana	
Laura Grabowski	Danilo Pianini	Porfirio Silva	

## Organizing Committee

- René Doursat (General Chair) *Complex Systems Institute, Paris Ile-de-France, CNRS – CREA, Ecole Polytechnique & CNRS, Paris*
- Hugues Bersini (Co-Founder) *IRIDIA, Université Libre de Bruxelles*
- Paul Bourguine (Co-Founder) *Complex Systems Institute, Paris Ile-de-France, CNRS – CREA, Ecole Polytechnique & CNRS, Paris*
- Tom Lenaerts (Program Chair) *MLG, Université Libre de Bruxelles – AI-lab, Vrije Universiteit Brussel*
- Mario Giacobini (Program Chair) *Dept. of Animal Productions, Epidemiology and Ecology – Molecular Biotechnology Center, University of Torino*
- Marco Dorigo (Co-Organizer) *IRIDIA, Université Libre de Bruxelles*

## Administrative and Onsite Staff

Finally, we want to thank for their invaluable help all those who were involved in the preparation of the event, in particular

- Marcel Skrobek (office head)
- Stéphanie Bayou-Zaba (assistant)
- Geneviève Tual (office manager)
- Razvan Dordea (MSc student) *Complex Systems Institute, Paris Ile-de-France, CNRS – CREA, Ecole Polytechnique & CNRS, Paris*
- Muriel Decreton (office manager)
- Eliseo Ferrante (PhD student)
- Vito Trianni (postdoc) *IRIDIA, Université Libre de Bruxelles*
- Manuele Brambilla (PhD student)
- Alessandro Stranieri (PhD student)
- Matthew Egbert (research fellow) *University of Sussex*
- Luca Ferreri (PhD student) *University of Torino*
- Regina Frei (postdoc) *Imperial College London*
- Carlos Quintana (PhD student) *University of Malaga*
- Gabriel Varela (student) *Bowdoin College, Brunswick, Maine*

## Art Exhibit: Graphic & Electronic Engineers, Set Designers

- François Murre (founder) *3D from Mars*
- Patrice Bersani (CEO) *DreamLabs*
- Delphine Fabbri-Lawson (art curator)

## Support

We are also very grateful to the following institutions for their financial support:

- Complex Systems Institute, Paris Ile-de-France, itself sponsored by the Région Ile-de-France
- CREA, Ecole Polytechnique & CNRS, Paris
- IRIDIA, Université Libre de Bruxelles
- Fondation Jean-Marie Delwart, Bruxelles

The organizing committee of ECAL'11:

René Doursat, Hugues Bersini, Paul Bourguine,  
Tom Lenaerts, Mario Giacobini, Marco Dorigo,  
*Paris, August 2011*

# General architecture of a genetic regulatory network. Applications to embryologic control

Jacques Demongeot<sup>1</sup>, Alexandra Henrion-Caude<sup>2</sup>, Athanase Lontos<sup>3</sup>, Emmanuel Promayon<sup>3</sup>

<sup>1</sup>Université J. Fourier Grenoble 1, AGIM, CNRS FRE 3405, 38706 La Tronche, France

<sup>2</sup>Université Paris Descartes, INSERM U 781, Hôpital Necker – Enfants Malades, 149 rue de Sèvres, 75015 Paris, France

<sup>3</sup>Université J. Fourier Grenoble 1, TIMC-IMAG, CNRS UMR 5525, 38706 La Tronche, France

Jacques.Demongeot@agim.eu

## Abstract

The general architecture of a genetic regulatory network consists of strong connected components of its interaction graph, to which are attached three kinds of sub-structures:

- a set of up-trees, rooted in the sources of the interaction graph, represented either by small RNAs like microRNAs: nuclear miRs or mitochondrial mitomiRs, i.e., translational inhibitors respectively of the messenger mRNAs and of the transfer tRNAs, or by gene repressors and/or inducers,
- a set of circuits in the core (in graph sense) of the strong connected components of the interaction graph,
- a set of down-trees going to the sinks of the interaction graph, i.e., to genes controlled, but not controlling any other gene.

The various state configurations it is possible to observe in the above sub-structures correspond to different dynamical asymptotic behaviors. The network dynamics have in general a small number of attractors, corresponding in the Delbrück's paradigm to the functions of the tissue they represent. Examples of such dynamics will be given in embryology: cell proliferation control network in mammals and gastrulation control network in *Drosophila melanogaster*.

## Introduction

Genetic networks can be considered as the analogues of neural networks for controlling the expression of genes. Their time constants are different (e.g., the rhythms of protein expression are of the order of magnitude of some minutes and those of neural firing are of some milliseconds) but their connectivity is about the same (in-degree between 1.5 and 3, i.e., the mean number of the genes or neurons influencing positively or negatively other ones is between 1.5 and 3) as well as the number of their strong connected components (rarely more than 2 for the control of a dedicated function). For these reasons, many common mathematical features have been adopted by the modelers in charge of designing the interaction graph of such networks: i) Boolean representation of the state space (1 if the gene is expressed, 0 if not), ii) Hopfield-like transition function (Demongeot and Sené, 2008d; Demongeot et al., 2008c, 2009b, 2011b, in press) and iii) extraction of the same features, like entropy and motifs (Demongeot et al., 2010). We will use in this paper this common theoretical framework in order to interpret examples of the genetic network dynamics.

## Generalities about the architecture of the interaction graph of a genetic network

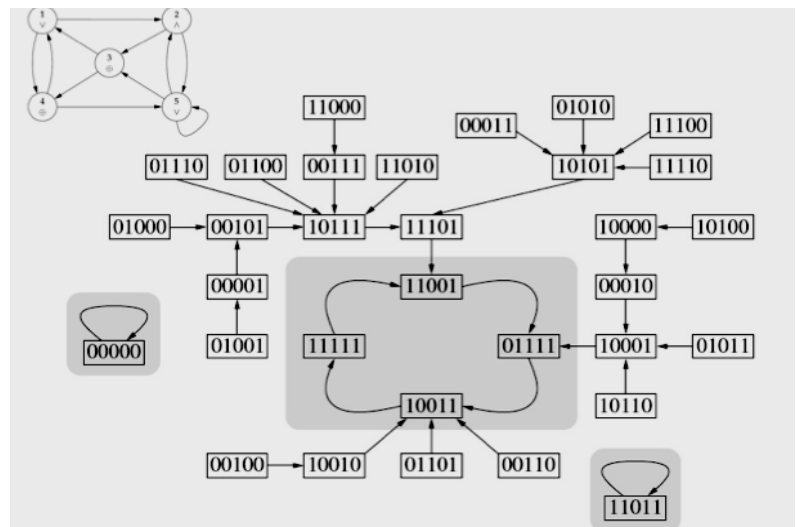


Figure 1: The interaction graph (top left) and the trajectory graph of a Boolean genetic regulatory network

The architecture of a genetic network can be decomposed into 3 directed graphs: i) the interaction graph with positive (resp. negative) arrows for induction (resp. repression), ii) the trajectory graph made of the consecutive states from an initial state until an asymptotic behavior (fixed state or limit-cycle of periodic states) and iii) the updating graph with an arrow between two genes if the target gene is updated after the source one. The knowledge about the first graph is given by DNA-protein interactions, about the second by DNA array devices recording gene expression and about the third by the chromatin clock. This architecture shows in Figures 2 and 3 some common features: i) a set of up-trees, issued from the sources of the interaction graph of the network, made either of small RNAs like siRNAs or microRNAs (nuclear miRs or mitochondrial mitomiRs, respectively translational inhibitors of the messenger mRNAs and of the transfer tRNAs), or of gene repressors and/or inducers, self-expressed without any

other gene controlling them, ii) a set of circuits in the core (in the graph sense) of the strong connected components of the interaction graph. These circuits are unique or multiple, disjoint or intersected, reduced to one gene or made of several ones, negative (having an odd number of negative interactions) or positive, and iii) a set of down-trees going to the sinks of the interaction graph.

By identifying each function of a regulatory network to one of the attractor of its dynamics as suggested by Delbrück (Demongeot, 1998), it is possible to count the number of the attractors provided by isolated circuits, and the number – largely reduced – brought by tangential or intersected circuits (Demongeot et al., 2009b, 2011a, 2011b, in press), depending on the updating mode fixed by the chromatin dynamics.

### The control of the genetic networks by microRNAs (miRs). Example of mitomiRs

Since a decade, numerous small RNAs issued from the non coding part of plant and animal genomes (like silencing siRNAs and microRNAs or miRs) have been found as inhibiting the translation by hybridizing the mRNAs with the help of RNA-binding oligo-peptides. This inhibition is partly aspecific because of the large number of possible mRNA targets for each small RNA. On Figure 2, the dynamics of a circuit of size 3 (3-circuit) is analyzed, when one gene of the 3-circuit is inhibited by a miR. If the inhibition is associated to another inhibition of this gene or if it is sufficiently strong, it is able to transform a limit-cycle behavior in a fixed configuration, the circuit being either negative or positive (Figure 2 top left and top middle). When the miR inhibition is less than the activation on the target gene, then the periodic behavior is conserved (Figure 2 top right). We can say that the inhibitory influence by the small RNAs is exerted only on sufficiently “weak” circuits, like on the carved (weak) zones of an etching on which only the nitric acid can carve.

Recently some nuclear miRs like miR-1977 (Figure 3) have been discovered whose targets are mitochondrial mRNAs coding for enzymes of the oxidative phosphorylation (Bandiera et al., 2011). Such miRs have been called mitomiRs (Dass et al., 2010). This discovery invited to examine if there exist parts of the non-coding mitochondrial DNA (called the d-loop, cf. Figure 3) susceptible to code for hybridizing RNAs blocking the free parts (the loops) of the mitochondrial tRNAs: the corresponding inhibition would be totally aspecific and exerted in situ without nuclear control in order to slow oxidative phosphorylation in absence of a strong energetic need. This effect could be useful for ruling the balance Pasteur/Warburg effect versus OxPhos effect, allowing to avoid both cancers in case of Pasteur/Warburg dominance and degenerative diseases in case of oxidative phosphorylation dominance (Demetrius et al., 2010; Israel and Schwartz, 2011).

Several sequences corresponding to the tRNA loops – essentially the tRNA D-loop, but also Anticodon-loop and TψC-loop have been found both in nuclear and in mitochondrial miRs. We will take in the following as reference the Lewin’s tRNA given in (Krebs et al., 2009): it

has been proved that the loops sequence in this reference tRNA was the closest among all known tRNAs to an Archetypal Basic RNA sequence of 22 bases (called RNA AB) verifying the following variational min-max principle:

- to be as short as possible,
- to present one and only one triplet corresponding to each amino-acid, in order to serve as “matrimonial agency” favouring the vicinity of any couple of amino-acids, close to RNA AB, and able to form strong peptide bonds (i.e., covalent chemical bonds formed by two amino-acids, when the carboxyl group of one reacts with the amine group of the other) between them, in order to initiate the peptide building as an ancestral tRNA, well conserved for example in the present Gly-tRNA of *Oenothera lamarckiana*.

For satisfying the constraints above, the RNA AB must be circular and contain at least 20 triplets. The minimal solution is given in (Demongeot and Besson, 1983; Demongeot and Moreira 2007; Demongeot et al. 2006, 2008a, 2009a, 2009c). The corresponding RNA AB sequence can be given in circular or hair-pin form and could be considered as the ancestor of the present tRNA loops. We will indicate in the following in blue the possible hybridization sites, by using the complementary pairing A-U, C-G and G-U:

1) for the nuclear mitomiRs, we have a pairing with:

- the D-loop and TψC-loop (13/22) (Bandiera et al., 2011)

**5' UAAAUUGGUACUGCCAUAAGA 3' AB**

**3' AAUUGUCGAUUCGUGGGAUUAG 5' miR 1977**

- the Anticodon-loop (12/22) (Bandiera et al., 2011)

**5' UUCAAGAUAAAUGGUACUGCCA 3' AB**

**3' AUAAGAGCGUGCCUGAUGUUGGU 5' miR 1974**

- the TψC-loop (12/22) (Bandiera et al., 2011)

**5' GAUAAAUGGUACUGCCAUAUAA 3' AB**

**3' AUCUUCCGAUCCUGGUUUGG 5' miR 1978**

2) for the mitochondrial mitomiRs, we have a pairing with:

- the D-loop (Cui et al., 2007)

the sequence **AAUGGUA** is found in many species in the CSB part of the mitochondrial d-loop (Figure 3)

- the TψC-loop (Sbisa et al., 1997)

the sequence **GUACAUU** is found in many species in the ETAS part of the mitochondrial d-loop (Figure 3)

Each pairing described above corresponds to a probability less than  $10^{-4}$  to occur (Demongeot and Moreira, 2007) and could correspond to the relics of an ancient protein building mechanism without ribosomes, in which the amino-acids were directly linked to RNA chains or cycles playing the role of matrimonial agency, i.e., facilitating the grouping of amino-acids, hence favoring the constitution of peptidic bonds between them (for other hypotheses concerning the catalysis of peptidic synthesis, see (Huber and Wächtershäuser, 1998; Hsiao et al., 2009)). When tRNA loops are hybridized by nuclear or mitochondrial mitomiRs, efficacy and specificity of the complex made of amino-acid, tRNA and amino-acyl-synthetase (enzyme esterifying an amino-acid for complexing it to a specific tRNA) can be affected, causing an inhibition of the translation mechanism.

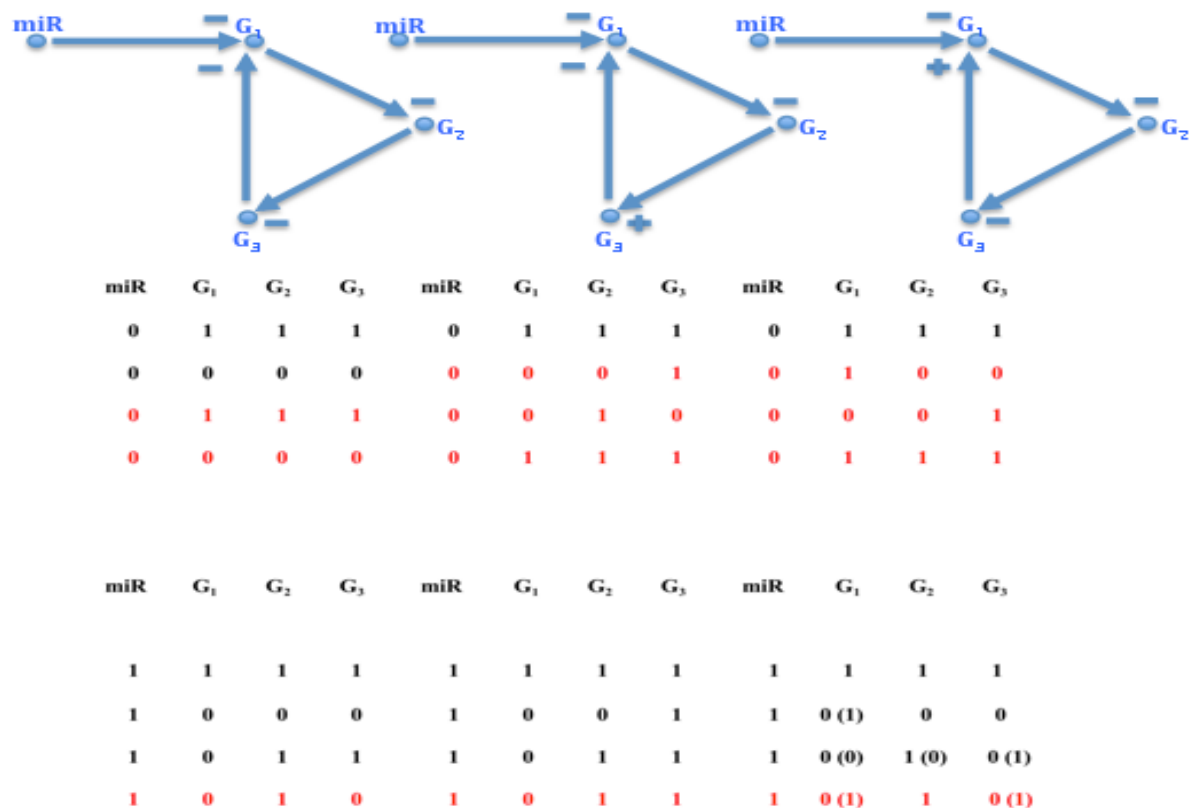


Figure 2: Top) Architecture of 3-circuits controlled by a miR, with negative (left) and positive (middle and right) circuits. Middle) Periodic dynamics when the miR is not expressed (miR=0). Bottom) Fixed configuration if the miR is expressed (miR=1), except if the miR inhibition is less than the gene activation (in parentheses), case in which the periodic behavior is conserved.

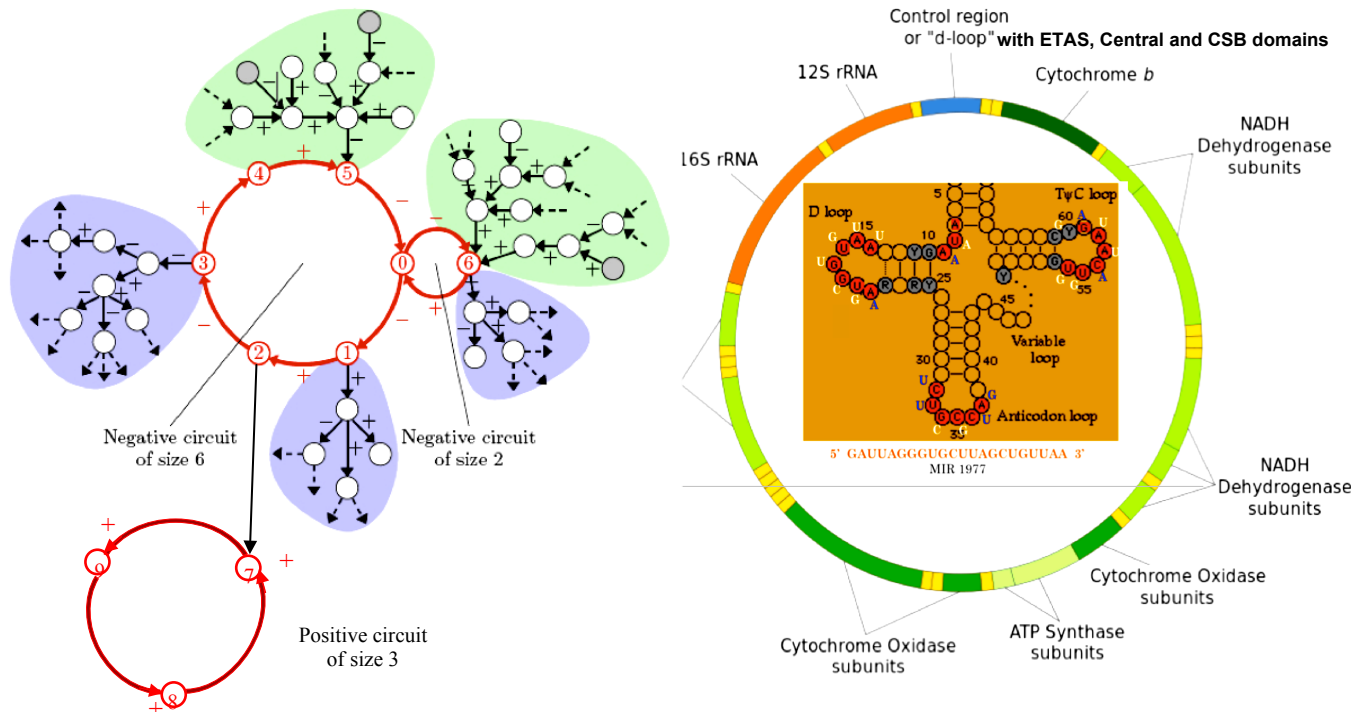


Figure 3: Left) General architecture of a genetic network with 2 circuit layers (red), 2 up-trees (green) and 3 down-trees (violet). Right) the circular mitochondrial DNA with its non-coding part (d-loop blue) and inside a tRNA structure hybridized by miR 1977.



# Genetic network ruling the cell-cycle

The genetic network ruling the cell-cycle in mammals, centered on the gene E2F, is crucial for cells because of its links with Engrailed network controlling: i) through gene Elk

the potassium channels in hippocampus neurons ruling the memory (Top of the Figure 4) and ii) through genes Engrailed/GATA-6, c-Myc and RAS, in a double incoherent control pathway (with both positive and negative arrows, respectively in red and green in Figure 4), the apoptosis and proliferation processes. This last control must be very precise if the tissue controlled has to keep constant its cell number. A way to obtain this acute control is to intersect in the Engrailed network several circuits (cf. Figure 4 Bottom right and (Demongeot et al., 2009b, 2011a, 2011b, in press)) and to exert an inhibitory control through miRs and/or mitomiRs, themselves possibly controlled by p53 (Figure 4 Middle).

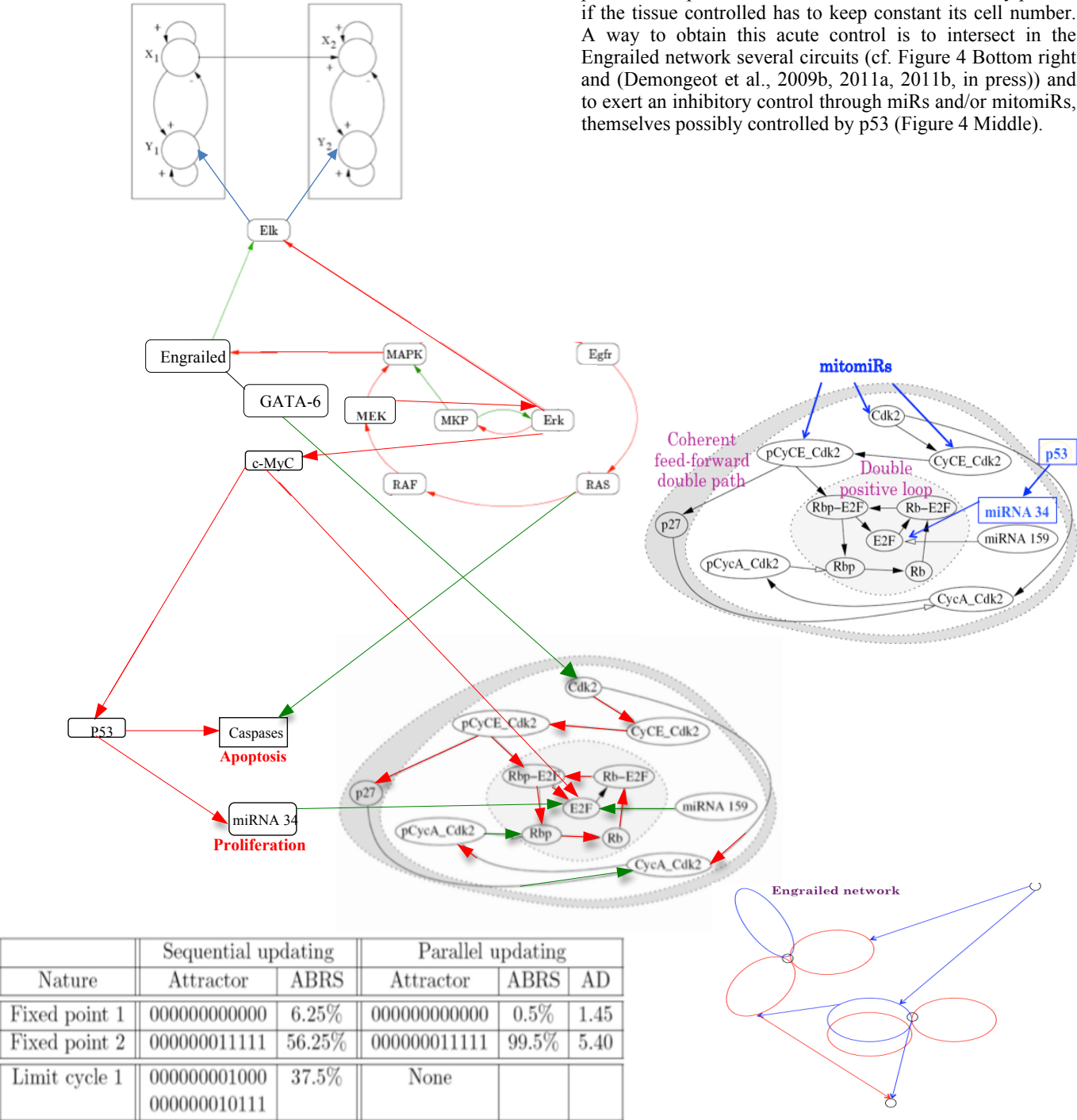


Figure 4: Middle right) Cell cycle controlling genetic network centered in mammals on the E2F box inhibited by small RNAs (miRs or nuclear and/or mitochondrial mitomiRs). Top left) Engrailed network controlling the potassium channels of hippocampus neural networks. Middle left) Engrailed network controlling both apoptosis and proliferation processes. Bottom left) Attractors of the dynamics specific to the E2F box. Bottom right) General structure of the Engrailed network.

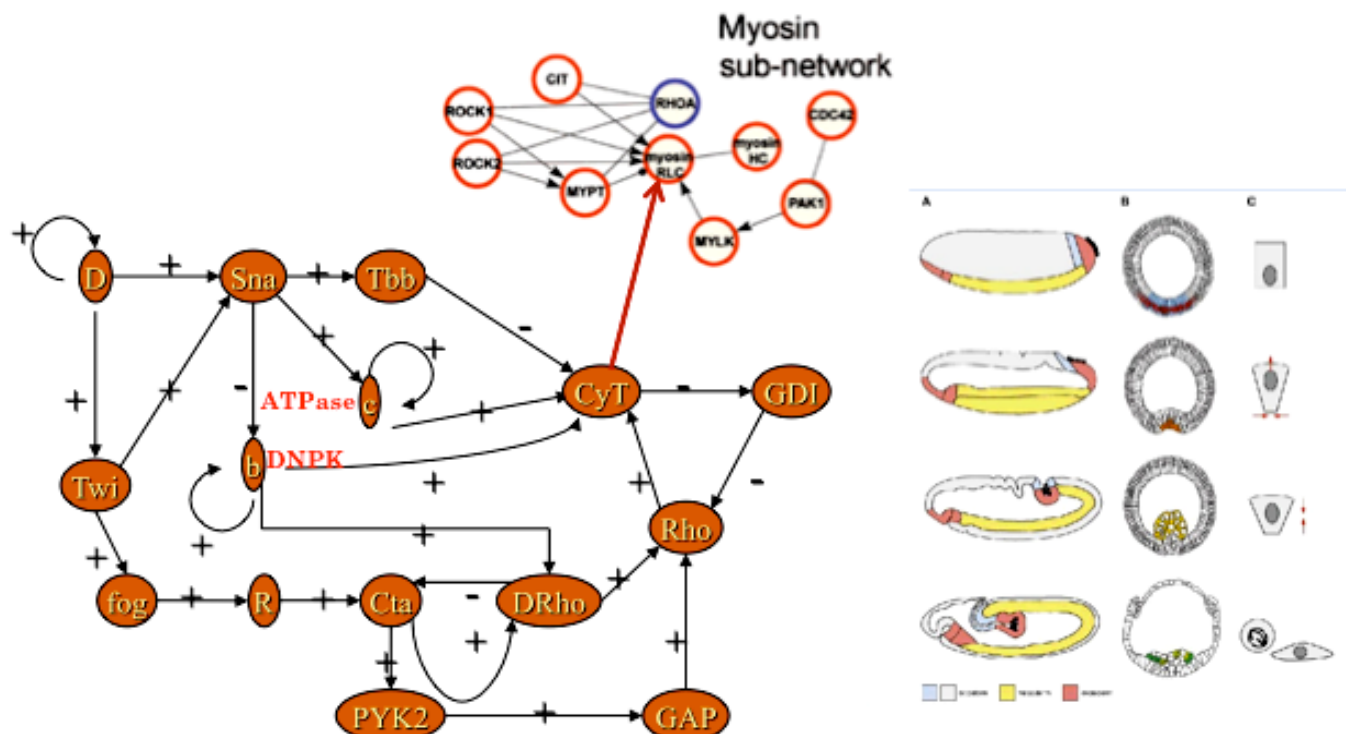


Figure 5: Bottom left) Gastrulation controlling genetic network from (Leptin, 1999) with addition of 2 ATP and GTP controlling enzymes b and c. Top middle) Myosin controlling subnetwork. Bottom right) The 4 differentiated cells needed for building the future digestive tube.

A triple action (accelerate, stop and slow down the cell cycle) on proliferation process is exerted negatively by the gene GATA-6 which is inhibited 1 time out of 2 by MAPK, and successively positively and negatively by the gene c-MyC which is activated 1 time out of 2 by Erk. The limit cycle of order 4 brought by the negative circuit of size 2 (MKP/Erk) leads genes MKP, Erk, MAPK, Engrailed, GATA-6, c-MyC, p53, miRNA34, Cdk2, E2F and caspases to the limit-cycle: 01100001001, 11110100001, 10011110000, 00000011011. Then the second fixed point of the E2F box is reached 1 time out of 4 and the caspases/apoptose box is activated 1 time out of 2: this result allows the exponential growth of the cell number to be compensated in a tissue by the linear growth of the apoptosis, 2 daughter cells replacing 2 dead cells during one period of the limit cycle, hence ensuring the conservation of the tissue volume and tissue function, any disequilibrium of the balance giving either a tumor growth or tissue rarefaction.

### Genetic network ruling the gastrulation

The gastrulation is a dynamical process occurring at the end of the blastula phase. It is an early embryonic stage, including mass movement of cells to form complex structures from a simple starting form. Experiments in vivo have shown that there are many types of mass cell movement taking place during gastrulation: ingression, invagination, involution, epiboly, intercalation and convergent extension. In the next Section, we will focus on the simulation of the phenomenon of invagination of cells, which leads to the creation of the ventral furrow. In order to control the gastrulation process, a genetic regulatory network has been proposed in (Leptin, 1999). This network has been improved by adding 2 genes

(Figure 5 Bottom left): ATPase (enzyme located inside the inner mitochondrial membrane ensuring the resourcing of ATP from ADP) and DiNucleotide Phosphate Kinase (enzyme resourcing GTP from GDP and ATP). This addition of genes allows the network to pass from 2 to 4 attractors, providing the 4 types of differentiated cells (from bottle cell to intestinal epithelial cell) needed to achieve and finish the digestive tube (Figure 5 Bottom right). The CyT node correspond to the genes involved in the CyToskeleton formation, i.e., essentially the genes of Actin, Tubulin and Myosin, the latter being controlled by a specific subnetwork (Figure 5 Top middle). When the genes coding for the two types of Myosin (RLC, with Regulatory Light Chain and HC, with Heavy Chain) are expressed, then the ventral furrow invagination can start. We will model this process in the next Section showing with a simple mechanical model that it begins by a cell contraction followed by an invagination at the two extremities of the Drosophila embryo, extended after to central embryo region.

### Physical Model of Ventral Furrow

Several successful models have already been created in order to simulate the process of ventral furrow invagination in *Drosophila melanogaster*. Although they have been extensively monitored, the parameters driving the movement and deformation of cells are not fully explained. We shall describe the structure of our physical model, the parameters we used to create it, the assumptions we made and the new possibilities and questions raised by this approach. This work focuses on the area of the structure where the phenomenon begins. As a result, we have modelled the upper part of one side of blastula (Figure 6) as described in (Abbas et al., 2009).

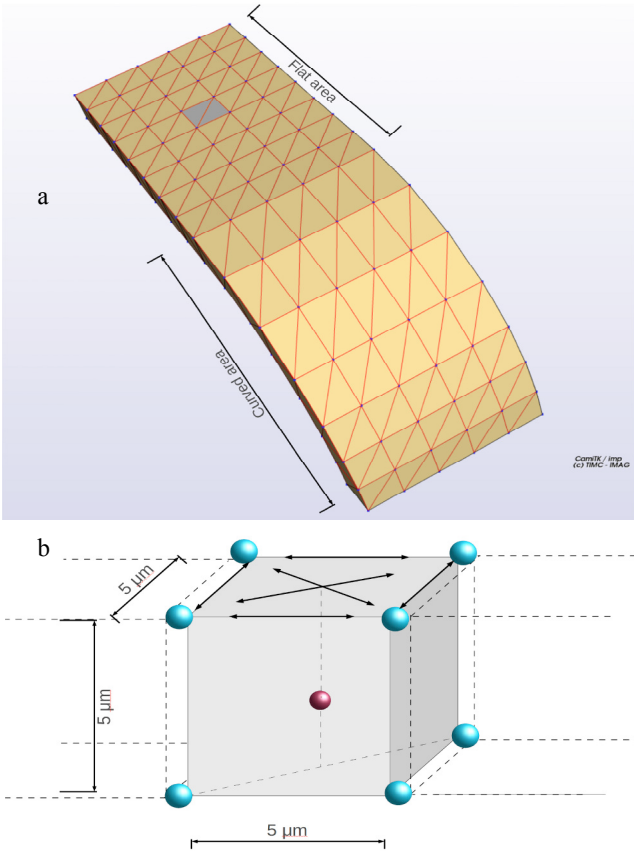


Figure 6: a) Representation of the simulated embryo structure at its initial shape and b) of an individual cell located at area of the centre row of the structure with its centrosome in red.

In our approach, the structure consists of 75 cells arranged in 15 columns of 5 cells each. The first 8 columns form the central part of the structure. The curvature of the structure starts at column 9 and ends at column 15, for a total curvature of  $90^\circ$  (Figure 6a). Each cell is modelled as a hexahedral object, composed of 9 particles. 8 particles are used as the vertices of the hexahedron and one particle is located in the middle, denoting the centrosome of each cell. The cells, with the aid of a biomechanical library, are defined as individual physical objects, with three distinct characteristics: incompressibility, elasticity and contractility. The structure is represented on Figure 6 at its initial shape and an individual cell is located at the central row of the structure. The grey cell corresponds to the cell presented in Figure 6b. The cells of the central area are modelled by cubes with edges of  $5\mu\text{m}$ , resulting to 6 facets of initial surface equal to  $25\mu\text{m}^2$ . The initial volume is  $125\mu\text{m}^3$ . Muscular forces (black arrows) connect the particles of the top facet of the cell. The red sphere represents the centrosome, initially located at the centre of the cell. The particles are modelled as nodes with the ability to interact with their environment. They are defined by their position and their mass. Elastic and muscular forces are applied on them and they can also be submitted to boundary conditions. Their combined displacement is the crucial factor that affects the cell deformation and movement. The

incompressibility algorithm, uses the facets geometry and a displacement constraint, to keep the volume of cells constant. Elasticity forces are defined between neighbouring particles in order to model the tissue reaction against deformation (Henon et al., 1999; Promayon et al., 2003). The elasticity parameters have a small value, so that the cell shape can be modified quite easily by other forces. As a result, we have deformable cells, with nearly unchangeable volumes (which imitates the behaviour of cells *in vivo*). In addition, using muscular forces, we can induce the contraction of cellular objects similar to those due to the Myosin excess (Patwari and Lee, 2008). Using a higher value of the elasticity parameter for the centre particle (centrosome), we ensured that this particle stays close to the centre of the cell, even when the cell is deformed. This allows us to model the rigidifying effect of the cytoskeleton. *In vivo* experiments have shown that neighbouring cells form Adherens Junctions (AJs), which contain complexes of the transmembrane adhesion molecule E-cadherin and the adaptors  $\alpha$ -catenin and  $\beta$ -catenin (Gumbiner, 2005; Martin et al., 2010). In addition, these AJs are formed in the apical areas of the lateral surfaces of the cells (Tepass and Hartenstein, 1994; Oda and Tsukita, 2000). In our model, we have considered AJs to offer very strong linking between cells. Therefore, the vertices of the hexahedron are merged, summing up the forces and constraints of all concurrently surrounding cells. This allows a faster propagation of the forces during the simulation.

## Simulation

Particles at the top of each cell in the central row are linked by muscular forces, which are used to model the forces applied by the orthogonal perpendicular Myosin fibres (Figure 6b). The norm of these forces for each particle is the same, resulting from a uniform distribution of forces along the structure, as suggested in (Brodland et al., 2010). More, boundary conditions are applied to the movement of some particles to verify the symmetry of the simulation (Figure 7):

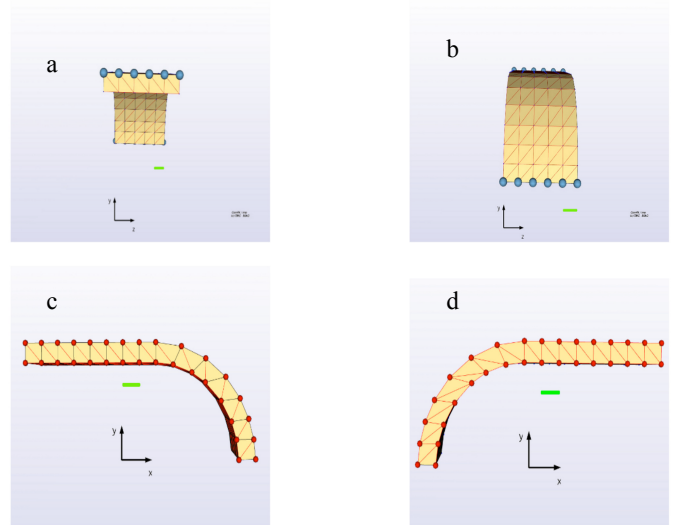


Figure 7: Representation of the boundary conditions imposed on the simulated embryo structure.



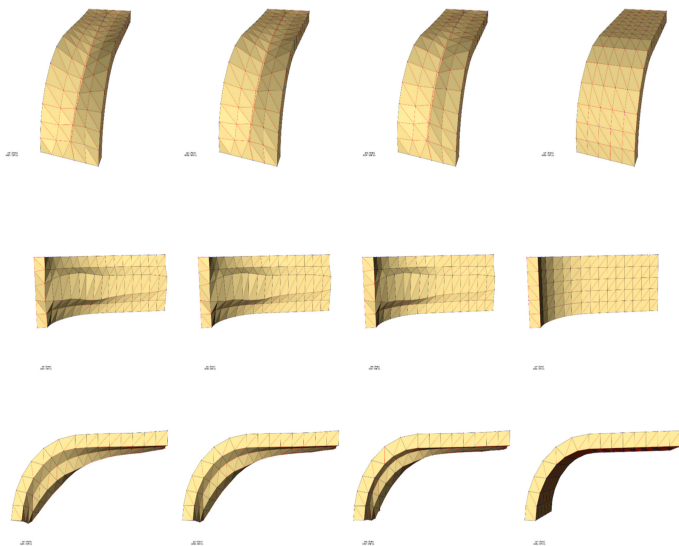


Figure 8: Simulation of the ventral furrow invagination process in *Drosophila melanogaster*.

i) the first boundary condition implies that structure edges cannot move in any direction (Figures 7a,b), ii) the second is applied on the side parts of the structure (Figures 7c,d).

The particles can “slide” on the x and y axis but they cannot move on the z axis. These boundary conditions allow the simulation to consider that this model is a part (Figure 8) of a bigger structure, with cells expanding from all sides, in order to form a tubular shape, as presented in Figure 9. At the beginning of the simulation, all the particles are submitted to forces of equal value. This is achieved by applying uniform elasticity and contractility coefficients along the structure. The simulation is divided in time-steps. Each time-step corresponds approximately to 0.05 seconds. At each time-step, the following processing takes place:

- the forces are summed up on all the particles and integrated along the structure using a classical integration scheme,
- the velocity and position of each particle are calculated and integrated also along the structure,
- the constraints are applied (incompressibility and boundary conditions).

In Figure 8, we present the geometry obtained for four different instances of the simulation, from three different angles. In the first row, the geometry is shown from the top, in the second row, it is shown from the bottom and in the third row it is shown from the side of the structure. In next papers to appear, we will provide videos of the entire simulation from all three points of view. At the beginning the cells in the centre row are contracting due to the activation of the Myosin fibres (after entering in the Bottle cells attractor of the previous Section). This contraction pulls all the cells of the model towards the centre. Due to the initial geometry of the structure, as shown in Figure 8, the vertical component of the force applied on the particles of the curved area causes the particles to move downward. As the cells located on the centre

row of the curved part move downward, they concurrently pull the other cells of the structure as well, due to the cell-cell bonds. As a result, all the cells start to move downwards (see Figure 8).

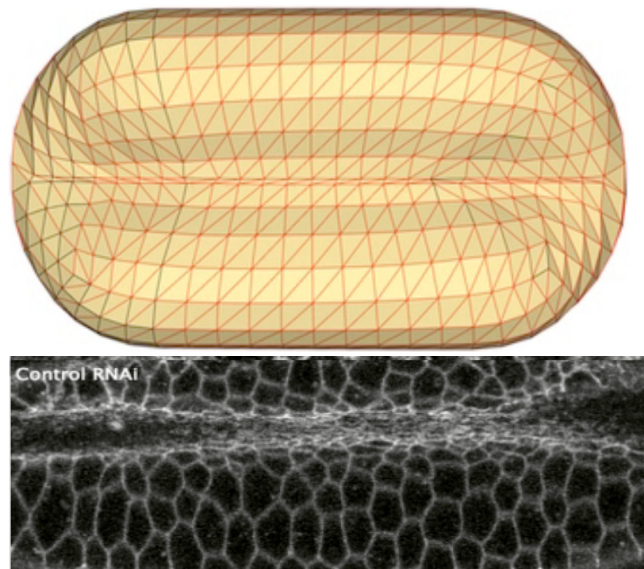


Figure 9: Simulation of invagination starting at the *Drosophila* embryo extremities (Bottom from (Martin et al., 2010)).

An important factor concerning the invagination process is the surface/volume ratio. In vivo experiments have shown that, as the phenomenon proceeds, the area of the cell in contact with the nourishment fluid decreases (Leptin, 1999). On the other hand, cell volume increases. As a result, the surface/volume ratio decreases with time. It has been noted that it can decrease up to a certain threshold, after which the cell tends to divide (Figure 9) as observed in (Cui et al., 2005).

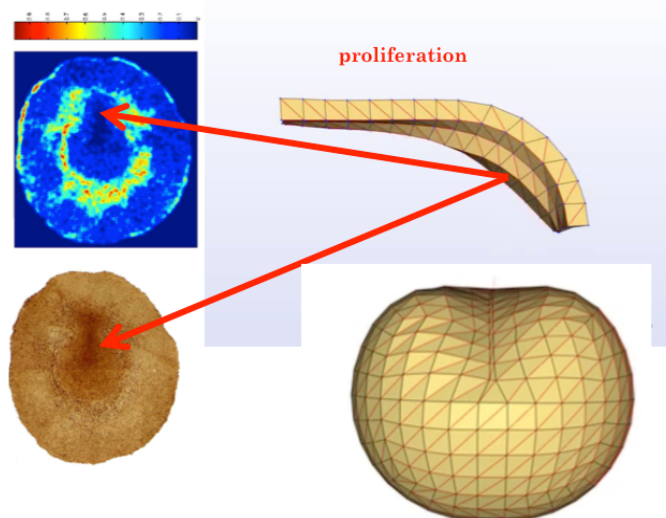


Figure 9: Proliferation occurring at the most invaginated part of the *Drosophila* embryo extremities, the Top left showing a BrDU pre-mitotic S-phase activity from (Cui et al., 2005)).

## Conclusion

We have shown in this paper that the general architecture of a genetic regulatory network involves several genetic circuits, which are crucial for imposing a dynamics having only few attractors, corresponding to few functions to fulfil. This small number of attractors is well controlled by the existence of circuit intersections as well as by the presence of an aspecific inhibitory “noise” from small RNAs, like miRs and mitomiRs.

## References

- Abbas, L., Demongeot, J. and Glade, N. (2009). Synchrony in Reaction-diffusion models of morphogenesis: applications to curvature-dependent proliferation and zero-diffusion front waves. *Phil. Trans. Royal Soc. A*, 367:4829-4862.
- Bandiera S, Rüberg S, Girard M, Cagnard N, Hanein S, Chrétien, D., Munnich, A., Lyonnet, S. and Henrion-Caupe, A. (2011) Nuclear Outsourcing of RNA Interference Components to Human Mitochondria. *PLoS ONE* 6:e20746.
- Ben Amor, H., Demongeot, J., Elena, A. and Sené, S. (2008). Structural Sensitivity of Neural and Genetic Networks. *Lecture Notes in Computer Science*, 5317:973-986.
- Brodland, G., Conte, V., Cranston, P., Veldhuis, J., Narasimhan, S., Hutson, M. S., Jacinto, A., Ulrich, F., Baum, B. and Miodownik, M. (2010). Video force microscopy reveals the mechanics of ventral furrow invagination in drosophila. *Proceedings of the National Academy of Sciences USA* 107:22111-22116.
- Cui, C., Yang, X., Chuai, M., Glazier, J. A. and Weijer, C. J. (2005). Analysis of tissue flow patterns during primitive streak formation in the chick embryo. *Developmental Biology* 284 :37-47.
- Cui, C., Ji, R., Ding, F., Qi, D., Gao, H., Meng, H., Yu, J., Hu, S. and Zhang, H. (2007). A complete mitochondrial genome sequence of the wild two-humped camel (*Camelus bactrianus ferus*): an evolutionary history of camelidae *BMC Genomics* 8:241
- Das, S., Ferlito, M., Wang, R., Liu, D., Raghavachari, N., Munson, P., Murphy, E. and Steenbergen, C. (2010). Existence of microRNA, “mito-miR-181c”, in Heart-Derived Mitochondria: Role in regulation of mitochondrial function by targeting mt-COX1. *Circulation* 122:A20624.
- Demetrius, L., Coy, J. F. and Tuszyński, J. A. (2010). Cancer prokifeartion and therapy: the Warburg effect and quantum metabolism. *J. R. Soc. Interface* 7:2.
- Demongeot, J. and Besson, J. (1983). Code génétique et codes à enchaînement I. *C.R. Acad. Sc. Série III* 296:807-810.
- Demongeot, J. (1998). Multi-stationarity and cell differentiation. *J. Biol. Systems.*, 6:1-2.
- Demongeot, J., Bezy-Wendling, J., Mattes, J., Haigron, P., Glade, N. and Coatrieux, J.L. (2003). Multiscale modeling and imaging: the challenges of biocomplexity. *Proceedings of the IEEE Soc.*, 91:1723 – 1737.
- Demongeot, J., Glade, N., Hansen, O. and Moreira, A. (2007). An open issue: the inner mitochondrial membrane (IMM) as a free boundary problem. *Biochimie*, 89:1049-1057.
- Demongeot, J. and Moreira, A. (2007). A circular RNA at the origin of life. *J. Theor. Biol.*, 249:314-324.
- Demongeot, J., Glade, N. and Moreira, A. (2008a). Evolution and RNA relics. A Systems Biology view. *Acta Biotheoretica*, 56:5-25.
- Demongeot, J., Elena, A. and Sené, S. (2008b). Robustness in neural and genetic networks. *Acta Biotheoretica*, 56:27-49.
- Demongeot, J., Glade, N., Moreira, A. and Vial, L. (2009a). RNA relics and origin of life. *Int. J. Molecular Sciences*, 10:3420-3441.
- Demongeot, J., Ben Amor, H., Gillois, P., Noual, M. and Sené, S. (2009b). Robustness of regulatory networks. A Generic Approach with Applications at Different Levels: Physiologic, Metabolic and Genetic. *Int. J. Molecular Sciences*, 10:4437-4473.
- Demongeot, J., Drouet, E., Moreira, A., Rechoum, Y. and Sené, S. (2009c). Micro-RNAs: viral genome and robustness of the genes expression in host. *Phil. Trans. Royal Soc. A*, 367:4941-4965.
- Demongeot, J., Elena A., Noual, M., Sené, S. and Thuderoz, F. (2011a). "Immunetworks", attractors and intersecting circuits. *J. Theor. Biology*, 280:19-33.
- Demongeot, J., Elena A., Noual, M. and Sené, S. (2011b). Random Boolean Networks and Attractors of their Intersecting Circuits. In Barolli, L. et al., editors, *IEEE AINA' 11 & BLSMC' 11*, pages 483-487. IEEE Press, Piscataway.
- Demongeot, J., Noual, M. and Sené, S. (in press). Combinatorics of Boolean automata circuits dynamics. To appear in *Discrete Applied Mathematics*.
- Forest, L., Glade, N. and J. Demongeot, J. (2007). Liénard systems and potential-Hamiltonian decomposition. Applications. *C. R. Acad. Sci. Biologies*, 330:97-106.
- Forest, L. and Demongeot, J. (2008). A general formalism for tissue morphogenesis based on cellular dynamics and control system interactions. *Acta Biotheoretica*, 56:51-74.
- Gumbiner, B (2005). Regulation of cadherin-mediated adhesion in morphogenesis. *Nat. Rev. Mol. Cell Biol.* 6:622-634.
- Henon, S., Lenormand, G., Richert, A. and Gallet, F. (1999). A new determination of the shear modulus of the human erythrocyte membrane using optical tweezers. *Biophys. J.* 76:1145-1151.
- Hsiao, C. L., Mohan, S., Kalahar, B. K. and Williams, L. D. (2009). Peeling the Onion: Ribosomes are ancient molecular fossils. *Molecular Biology and Evolution* 26:2415-2425.
- Huber, C. and Wächtershäuser, G (1998) Peptides by activation of amino-acids with CO on (Ni,Fe)S surfaces: Implications for the origin of life. *Science* 281:670-672.
- Israel, M. and Schwartz, L. (2011). On the metabolic origin of cancer : substances that target tumor metabolism. *Biomedical Research* 22, 22 :132-166.
- Krebs, J. E., Kilpatrick, S. T. and Goldstein, E. S. (2009) *Lewin's Genes X*. Jones & Bartlett Learning, Sudbury, MA.
- Leptin, M. (1999). Gastrulation in drosophila: the logic and the cellular mechanisms. *EMBO J.* 18:3187-3192.
- Martin, A., Gelbart, M., Fernandez-Gonzalez, R., Kaschube, M., Wieschaus, E. (2010) Integration of contractile forces during tissue invagination. *J. Cell Biol.* 188:735-749.
- Michon, F., Forest, L., Collomb, E., Demongeot, J. and Dhouailly, D. (2008). BMP-2 and BMP-7 play antagonistic roles in feather induction. *Development*, 135:2797-2805.
- Oda, H. and Tsukita, S. (2000). Real-time imaging of cell-cell adherens junctions reveals that drosophila mesoderm invagination begins with two phases of apical constriction of cells. *Journal of Cell Science* 114:493-501.
- Patwari, P. and Lee, R. (2008) Mechanical control of tissue morphogenesis. *Circ. Res.* 103:234-243.
- Pratt, A. J. (2010). Evolution, Selection and the Metabolic Foundations of the RNA World. In Fellermann, H. et al., editors, *Artificial Life XII Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*, pages 49-56. MIT Press, Cambridge, MA.
- Promayon, E., Martiel, J. L. and Tracqui, P. (2003). Physically-based 3D simulations of cell deformations and migrations. In Alt, W., Chaplain, M., Griebel, M. and Lenz, J., editors, *Polymer and Cell Dynamics -Multiscale Modeling and Numerical Simulations*, pages 125-138, Birkhauser, Berlin.
- Sbisà, E., Tanzariello, F., Reyes, A., Pesole, G. and Saccone, C. (1997). Mammalian mitochondrial d-loop region structural analysis: identification of new conserved sequences and their functional and evolutionary implications. *Gene* 205:125-140.
- Tayyab, M., Lontos, A., Promayon, E. and Demongeot, J. (2011c). Modelling and image processing of constriction and proliferation in the gastrulation process of *Drosophila melanogaster*. In Barolli, L. et al., editors, *IEEE AINA' 11 & BLSMC' 11*, pages 473-477. IEEE Press, Piscataway.
- Tepass, U. and Hartenstein, V. (1994). The development of cellular junctions in the drosophila embryo. *Developmental Biology* 161:563-596.

# Can we Computerize an Elephant?

David Harel<sup>1</sup>

<sup>1</sup>Dept. of Computer Science and Applied Mathematics  
The Weizmann Institute of Science  
Israel  
dharel@weizmann.ac.il

## Abstract

The talk shows how techniques from computer science and software engineering can be applied beneficially to research in the life sciences. We discuss the idea of comprehensive and realistic modeling of biological systems, where we try to understand and analyze an entire system in detail, utilizing in the modeling effort all that is known about it. I will address the motivation for such modeling and the philosophy underlying the techniques for carrying it out, as well as the crucial question of when such models are to be deemed valid, or complete. The examples will be from among the biological modeling efforts my group has been involved in: T cell development, lymph node behavior, organogenesis of the pancreas, and fate determination in the reproductive system of the *Caenorhabditis elegans* nematode worm. The ultimate long-term “grand challenge” is to produce an interactive, dynamic, computerized model of an entire multi-cellular organism, such as the *C. elegans*, which is complex, but well-defined in terms of anatomy and genetics.

# Do We Need a Theory in the Era of Massive Data Flow?

Takashi Ikegami<sup>1</sup>

<sup>1</sup>Department of General Systems Sciences  
The Graduate School of Arts and Sciences  
University of Tokyo  
ikeg@sacl.c.u-tokyo.ac.jp

## Abstract

Massive Data Flow (MDF) is everywhere these days; from data about neural cells, social insects and genetic networks, to Lifelog (digital storage of a person's visual and audio life log) and SNS (Social network service) data streams. Current web and device technology has made it possible for us to record detailed and massive data flows of artificial and real living systems.

But how can we analyze and understand MDF? Can a simple toy model based on a plausible narrative and simulation still tell us something? Concepts like “the edge of chaos” and “self-organized criticality” once helped us to understand living systems, but we do not know whether the same concepts can be useful to MDF.

I think studies of artificial life in MDF need larger models, because we need the strength of models that overcomes MDF. Possible larger models do not have to mimic existing living creatures but can be larger, in the sense of novel invention and utilization of space and time. In other words, to understand the complexity of MDF is to recast and reconfigure it into a larger artificial model. Indeed, I myself made a large model called “MTM” (Mind Time Machine) in 2010 that ran for three months in an open space, receiving massive visual data from the environment with 15 cameras, processed by internal neural dynamics with a learning capability, and showing sustainable complex adaptive dynamics.

We need a theory to make large artificial life models and to take them out into the real world.

# Answering Descartes: Beyond Turing

Stuart Kauffman<sup>1</sup>

<sup>1</sup>Departments of Mathematics and Biochemistry

University of Vermont, USA

<sup>2</sup> Department of Signal Processing

Tampere University of Technology, Finland

stuart.kauffman@uvm.edu

## Foreword

Article reproduced with permission from a chapter in S. Barry Cooper and Andrew Hodges (editors): *The Once and Future Turing: Computing the World*, Cambridge University Press, to appear 2012.

## Introduction

The first half of the 20th Century was filled with a stunning group of scientists, Einstein, Bohr, von Neumann and others. Alan Turing ranks near the top of this group. I am honored to write in this Centennial Volume commemorating his work. How much do we owe one mind? His was a pivotal role in cracking the Nazi war code that profoundly aided the defeat of Nazism. His invention of the Turing machine has revolutionized modern society, from universal Turing machines to all digital computers and the IT revolution. His model of morphogenesis, the first example of a “dissipative structure”, to use Prigogine’s phrase for it, is one I have myself used as a developmental biologist.

I rightly praise Turing, but seek in this chapter to go beyond him. The core issue is the human mind. Two lines of thought, one stemming from Turing himself, the other from none other than Bertrand Russell, have led to the dominant view that the human mind arises as some kind of vast network of logic gates, or classical physics “consciousness neurons”, to use F. Crick’s phrase in *The Astonishing Hypothesis* (1), connected in the 10 to the 11th neurons of the human brain.

I think this view could be right, but is more likely to be wrong. My aim in this chapter is to sketch the lines of thought that lead to the standard view in computer science and much of neurobiology, note some of the philosophic claims for and doubts about the claim, but most importantly I wish to explore the emerging behavior of open quantum systems, their new physics, and, centrally, our capacity to construct what I will call non-algorithmic, non-determinate yet non-random Trans-Turing Systems. As we shall see, Trans-Turing systems are not determinate, for they inherit the indeterminism of their open quantum system aspects, yet non-random due to their classical aspects. They are new to us, and may move us decisively beyond the beauty but limitations of Turing’s justly famous, but purely classical physics, machine.

Beyond the above, I shall make one truly radical proposal that I believe grows out of Richard Feynman’s famous “sum over all possible histories” formulation of quantum mechanics,(2). This formulation is fully accepted as an equivalent formulation of quantum mechanics. I will show that Feynman’s formulation evades Aristotle’s Law of the Excluded Middle, while classical physics and, a fortiori, algorithmic discrete state, time, classical physics, Turing machines, obey the Law of the Excluded Middle. Following philosopher C.S. Pierce, who pointed out that “Possibles” evade the Law of the Excluded Middle, while Actuals and Probable obey that Law,(3), and Alfred North Whitehead,(4), I shall propose for our consideration a new dualism, Res potentia and Res extensa, the realms of the ontologically real Possible and ontologically real Actual, linked, hence truly united, by quantum measurement. In contrast, the dualism of Descartes, Res cogitans, thinking stuff, and Res extensa, his mechanistic world philosophy, have never been united. I believe Res potentia may be a consistent and new interpretation of “closed” quantum systems prior to measurement. These ideas and other much less radical ones resting on open quantum systems lead to new and testable hypotheses in molecular, cellular, and neurobiology, and, hopefully, a new line of ideas in the philosophy of mind including proposals about: how mind acts acausally on brain, an ontologically responsible free will, what consciousness IS, the experimentally testable loci of qualia as associated with quantum measurement itself, the irreducibility of both qualia and quantum measurement, the unity of consciousness, i.e. the “qualia binding problem” and its cognate “frame problem” in computer science. From these, technological advances in numbers of directions may flow.

## Mind as Machine

As noted, there are two strands, from Turing and the Turing machine, and from Bertrand Russell, that both lead to the view of the mind as a classical physics “computing machine”.

The strand from Turing is well known. It begins with the Turing machine, the very definition of algorithmic behavior. To recall, a Turing machine consists, in general, of an infinite tape divided into squares. On each square one of a finite number of more than one symbol, say “0” and “1”, is written. A reading head begins poised over one square. The head



contains two sets of rules. The first rule prescribes the following actions: If situated over a tape square with a given symbol written on it, the head will stay where it is, move one step to the left or right, erase the symbol on the square below it, and write a symbol from the defined alphabet of discretely different symbols on that square. The second rule specifies that under the above conditions, the reading head will change from one of a finite number of discrete internal states to some internal state. Thereafter, the system iterates. There is, in addition the crucial “halting state”.

Turing showed that any recursive computation that could be carried out could be carried out by a universal Turing machine. From this followed wonderful theorems about the formal undecidability of the “Halting problem”, the demonstration that most irrational numbers were not computable, and other remarkable advances.

The feature of the Turing machine I wish to emphasize is that it is *absolutely definite* or *determinate*. Given the symbols written on the tape, and rules in the reading head, its behavior at each step is fully determined. This determined behavior is essential to the algorithmic character of the Turing machine. Because it is determinate, the Turing machine is bound by classical physics. However, Turing machines are discrete state and discrete time systems, while classical physics more generally is based on continuous variables and continuous time and is also deterministic, and can, since Poincaré, exhibit deterministic chaos.

Computer scientists often distinguish between algorithms that may halt with an answer, and those that are “processes”, such as Holland’s Genetic Algorithm (5), which just continues or halts at some defined success criterion.

Turing, in the Turing Test, or “Imitation Game” (6), soon turned to the question of whether the human mind was itself a Turing machine. He thought, after careful consideration, that the answer was “Yes”. He did, however, retain doubts, partially reflected in his use of humans, not algorithms, as the judges in the Turing Test. Turing scholars rightly admire his capacity to doubt himself.

## Russell and Onwards to Mind as Machine

At the turn of the 20th Century, Bertrand Russell, having just published with Whitehead the *Principia Mathematica*, turned to the problem of maximally reliable knowledge of the “external world”. We could be wrong, he reasoned, that there was a chair in the room. But we could hardly be wrong that “We seemed to be seeing a chair”. That is, statements about our experiences, say visual, were less corrigible, or error-prone, than our statements about the external world. Russell and his contemporaries, including the young Ludwig Wittgenstein, hoped to build up knowledge of the external world from experience itself.

Pause and look at the room or world around you. You experience a “whole” visual field called in neuroscience, the “Unity of Consciousness”. This unity will be central to my interests. However, Russell threw away the Unity of Consciousness in his very first philosophic move. He invented, whole cloth, “Sense Data”, such as “Red here” or the musical note “A flat now” (7). That is, Russell shattered the unity of consciousness into bits, soon to be related to computational “bits”.

Russell’s next step was to invent “Sense data statements”. “It is true for Kauffman that ‘A flat now’”, (7).

Why did Russell make this move? Because his *Principia* hoped to construct the entire mathematical world from first order predicate calculus. Then the hope was that the statement, “There is a chair in the room”, could be translated into a *logically equivalent statement* comprised of a finite list of true or false sense data statements and quantifiers such as “There Exists”, and “For All”. If the move worked, knowledge of the external world would be set on a firm foundation.

The discussion took perhaps 40 years, but the move, culminating in the *Tractatus Logico-Philosophicus* by Wittgenstein (8), did not work. The statement, “There is a chair in the room” could not be translated into a logically equivalent set of sense data statements in the first order predicate calculus. Philosophers gave up on the idea that there was a “basement” language from which all other knowledge of the world, captured in propositions, could be formulated.

Famously, the later Wittgenstein, in his transforming opus *Philosophic Investigations* (9), pointed out that there was no basement language. Rather, language about legal proceedings could not be translated into logically equivalent sets of statements about ordinary human actions. Each “level” constituted a “language game”, not reducible to a lower level. Thus: “Kauffman is guilty of murder.” requires for its understanding a co-defined set of concepts such as “trail”, “jury”, “legally admissible evidence”, “legally competent to stand trial”...that cannot be translated or “reduced” into sets of statements about ordinary human actions.

This step is critical, for it says that there is no logical procedure, surely no first order logic, to get from a lower level language game, here normal human action, to a higher level language game, here legal language. But then there is no first order logic “algorithmic procedure” to get from the lower to higher language. Yet we learn legal language. This is one line of argument that the human mind is not merely algorithmic.

Despite some philosophers giving up on a basement language, the early cyberneticians, W. McCulloch and W. Pitts, in 1943 published a seminal paper that would lead to the contemporary theory of neural networks and “connectionism”.

McCulloch and Pitts showed that in a network of on/off formal neurons, constructed in a feed forward network, N formal neurons per row and M rows, and in which the input row “neurons” could be placed in any arbitrary combination of “1” and “0” states, the network, with arbitrary threshold Boolean functions such as *and*, *or*, and *not*, could compute any logical function on the “states” of the input neurons.

Implicitly, they identified the “1” or “0” state of a formal neuron with the truth or falseness of a Russellian sense data statement, such as, “For Kauffman, ‘A flat now’ is true”, which might be encoded by a “1” on the first neuron in the input layer to the feedforward network of formal neurons.

More generally McCulloch and Pitts considered networks with feedback loops.

They entitled their paper, “A logical calculus of the ideas immanent in nervous activity” (10).

In this step, McCulloch and Pitts set the stage for the now generally accepted view in computer science, neurobiology, and much of the philosophy of mind, that an “idea” in the

mind was logically identical to the on or/off states of a set of formal neurons, or in contemporary neurobiology, with the axonal firing or not of members of a set of “consciousness neurons”.

Note that McCulloch and Pitts chose the terms, “immanent in nervous activity”. In some magical way, the sense data features, or sense experiences, or “qualia” are “slipped” into the 1 and 0 behaviors of the formal neural net.

Note further that this conceptual move: 1) assumes that there is a basement language, captured in the “1” and “0” states of the formal neurons. 2) Has, with Russell, thrown away the Unity of Consciousness and will have to reconstruct it. In contemporary neurobiology this issue has returned as the famous “binding problem” ie how does the firing of unconnected “consciousness neurons” become bound into a unity of consciousness, or more simple examples such as this, from F. Crick’s *The Astonishing Hypothesis* (1): Suppose I see a yellow triangle and blue square. Suppose “yellow”, “triangle”, “blue” and “square” are, in fact, processed in different, unconnected, areas of the brain. How do “yellow” and “triangle” become bound together, while “blue” and “square” become bound together?

Following the logic of McCulloch and Pitts, the early hope was brain “grandmother cells” that fired if and only if you saw a combination of features, sense data, that equaled your grandmother. Now reconsider the number of relational features of your visual field. How many grandmother cells would be required, each to encode by firing “if and only if” presented with one of each of the possible combinations of up to, say 30 features at a time, out of say 10,000 features you can discriminate? The answer is  $(10,000 \text{ Choose } (30))$ , i.e.  $(10,000!) / 9,9770! \times (30!)$  a vast number. Crick (ibid), concludes that the idea does not work, it would take more than the 10 to the 11th neurons to encode all the sets of relational features you see.

One current hope is a 40 Hertz oscillation in the brain. The idea is that if “yellow” and “triangle” neurons fire at the same phase of the oscillation they will be bound, and if “blue” and “square” fire at a different phase, they too will be bound. Well, maybe, but how do we squeeze maybe trillions of combinations of relational degrees of freedom into different phases of a 40Hertz oscillation? I find it implausible. While detailed work on binding is beyond the scope of this chapter, in general the issue remains binding anatomically unconnected classical physics neurons and their presumed qualia, or experiences.

Note that this binding problem arises, descendant from Russell, with the idea of sense data and sense data statements, true or false, as a *digital and propositional encoding* of our experience of the world in our Unity of Consciousness. Below I will offer an unexpected *analog and non-propositional* encoding which may solve the binding problem.

But there is another deeper issue: McCulloch and Pitts, and all later neural network theory, cannot meet Wittgenstein’s language game argument that there is no “basement language” and the learning of higher language games cannot be based on algorithmic procedures from that basement language.

Despite the warning of Wittgenstein, connectionism has flowered, much along the ideas above, but with important improvements such as Back Propagation (11), and Hopfield Networks (12), with attractors encoding classes or memories

and content addressable memory. These are now the basis of voice recognitions systems around the world. But the language game problem remains unsolved, so mind seems not to be algorithmic on this ground.

I point to another important line of evidence that the mind is not algorithmic. I ask you to name all the possible uses of screwdrivers: screwing in screws, opening paint cans, tied to the end of a stick to spear fish, rented to locals to spear fish and you take 5% of the catch...Is there a statable list of the possible uses of a screwdriver for all possible purposes? I think not. How would we construct such a list? Know we had completed the list, or at least made it “infinite but recursively enumerable”? Yet we find new uses for screwdrivers and other artifacts all the time. This is the famous “frame” problem of computer science, never solved algorithmically. I believe there *is no bounded or recursively orderable set of functionalities of human artifacts for all possible purposes*, yet we literally discover and invent them all the time in the evolution of the econosphere. We routinely solve the frame problem. If so, the human mind is not always algorithmic.

I note that R. Penrose, in *The Emperor’s New Mind* (13), and *Shadows of the Mind* (14) also argues that the human mind is not always algorithmic based on its capacity to prove incompleteness theorems such as Godel’s theorem and the Halting Problem. I join Penrose, who precedes me, but on different grounds, in thinking the mind is not algorithmic and join him in thinking that quantum mechanics is related to consciousness.

## Mind, Consciousness, and the Mind as Machine

Two major positions can be taken with respect to mind as a classical physics, and further, a discrete space, time, and state algorithmic computational machine with inputs from a discrete space, time and state environment. First, we are not conscious at all, but are zombies. This view is discussed by Daniel Dennett in *Consciousness Explained* (15), which is, in part, a sophisticated form of logical behaviorism making use of an extensively developed computer science framework. A contrary argument is made by John Searle in his debated but famous Chinese Room argument which claims to show that mind is not a Turing machine, which is merely syntactic in its manipulation of symbols having no semantics, hence the Turing machine cannot experience the meanings of words (16).

In one form or another, the view of the mind-brain system as a network of classical physics neurons, with continuous variables, and continuous time, interacting in classical physics causal ways via action potentials, vast networks with classical physics inputs and outputs, is the dominant view today. Gerald Edelman, *Bright Air, Brilliant Fire* (17), Francis Crick, *The Astonishing Hypothesis* (1), John Searle, *The Mystery of Consciousness* (16), and most working neuroscientists hold this view. According to Searle, Functionalists such as H. Putnam and D. Lewis are “property dualists” who see mental terms such as “believe” as constituted by a classical physics causal network, whether made of neurons or beer cans. Searle asserts that functionalists do not mean by mental terms the actual experience of, for example, pain (16). These two

paragraphs cannot characterize the vast scholarly work above, yet these efforts neither answer Descartes, introduced just below, nor finds a home for consciousness itself.

Then whence consciousness, experience, qualia? A popular view is that at some level of complexity of a network of logic gates, whether electronic, water bowls pouring into one another above and below a 0/1 threshold, or classical physics continuous time and state neurons, consciousness will “emerge”. It is popular to point out that a single H<sub>2</sub>O molecule is not wet, but a sufficient collection of them is. So too, consciousness can emerge.

Perhaps consciousness can so emerge, but here is the first deep problem. If the emergent consciousness is a classical physical “process”, for example an electromagnetic field as some argue, then it is a deterministic classical physical system. Consider Newton’s three laws of motion and universal gravitation, and billiard balls moving on the table. The boundary conditions of the table and current positions, momenta and diameters of the balls entirely determine the entire future trajectory, perhaps deterministic chaos, of the sets of balls.

But if the mind-brain is a deterministic machine, we can have no ontologically real and responsible free will. I walk down the street, kill the little lady with a frying pan, but I am not responsible. I was physically determined to whack her. Even in the face of deterministic chaos I have no ontologically real responsible free will, merely perhaps the epistemic illusion of one.

Thus, the familiar view, derived from Turing and Russell, may be right, consciousness may be a classical physical “something”, but we buy it at the price of no ontologically real responsible free will.

It is a huge price to pay. I will offer below a set of ideas that appear to afford us, among other things, an ontologically responsible free will.

There is another huge set of problems, derived from Descartes in 1637 in his Discourse on Method. Descartes postulated a famous dualism (18): *Res cogitans*, thinking stuff, and *Res extensa*, his mechanical world view which led, a century later to Newton and celestial mechanics, and thence to classical physics.

But the problem immediately arose how *Res cogitans* is connected to *Res extensa*. Descartes proposed the pineal gland. The idea does not work.

Given Newton, here is the issue: If the brain is a deterministic dynamical system, like the billiard balls on the table, then the current state of the brain is entirely sufficient for the next state of the brain. Then there is *nothing for mind to do. worse, there is no way for mind - experiences - to act on brain!* What should mind do, some- magical-how cause the billiard balls to swerve despite the sufficiency of Newton’s laws?

This central problem arises due to the *causal closure of classical physics*. It is due to causal closure that we claim Newton’s laws, plus the initial positions and momenta and diameters of the billiard balls and boundary conditions, plus Newton’s laws in differential form, once integrated, are entirely sufficient to yield the entire future trajectory of the balls on the table.

Thus, the Turing model of the Machine Mind leaves us with no free will, and mind, experiences or qualia, if they can

arise at all, as unable to affect the classical physics machine aspect of the mind-brain system. We retreat to mind as a mere epiphenomenon, of no effect in our actions as humans, or a “compatibilism” which rejoices that at least as deterministic systems we can train one another to be moral machines.

In truth, we have been stuck with this cycle of problems since Descartes. Turing machine minds are frozen in the same way.

If the central problem above is due to the causal closure of classical physics, then I believe we must forsake the limitations of classical physics and purely classical physics “consciousness neurons” for a view that embraces the non-determinant behavior arising from quantum mechanics.

I turn now to such a radically different approach to the mind-body problem. It will take us through open quantum systems, the “Poised Realm” between open quantum and classicality for all practical purposes, FAPP, to non-determinate, hence non-algorithmic, yet non-random Trans Turing systems beyond Turing, to my tentative postulate about a new dualism, ontologically real *Res potentia*, the realm of the Possible, and *Res extensa*, the realm of the Actual, linked - hence united - by quantum measurement. This postulate is also an interpretation concerning what the unmeasured Schrodinger wave is “about”, where we have had no idea since the Schrodinger equation in 1927 (19). The postulate of *Res potentia* leads to a resulting idea of consciousness as a participation in *Res potentia*, ie in ontologically real Possibilities and strengthens the independent hypothesis that qualia, i.e. conscious experiences, are associated with quantum measurement. Most of what I shall say is independent of a real *Res potentia*.

But there is more: We escape the digital “propositional” model of mind with the realization that a quantum wave process in a potential well knows in an analog, not propositional or digital, way its potential well boundary condition or “context”, as part of solving the binding problem. I will link this analog “knowing of qualia” to quantum entanglement among many synapses in the brain as candidate loci of quantum behavior, and *quantum measurement* of those entangled degrees of freedom to achieve non-local EPR high correlations (20), hence “binding” of vastly many qualia, one per measurement, to solve the binding problem and achieve the Unity of Consciousness.

## Answering Descartes

With the discovery that chlorophyll wrapped by its chromophore bearing antenna protein can be quantum coherent for 700 femtoseconds or more,(21), “quantum biology” is emerging. I believe, however, that quantum coherence may be only a small part of quantum effects in biology. We biologists may find ourselves learning and collaborating with quantum physicists, and quantum chemists, in untellable ways. This part of the chapter is an attempt to see into this new territory.

## Closed Quantum Systems and the Two Slit Experiment

Many readers will be familiar with the famous two slit experiment (22). A photon gun emits photons, say one per minute, at a screen with two slits close together and behind the screen is a photodetector, say a film emulsion. If either slit is covered, one obtains a bright spot on the photodetector behind the open slit. Stunningly, if both slits are open, one obtains the famous bars of light, dark, light, dark..., the interference pattern. No classical objects, such as classical particles, can yield this result. It is the hallmark mystery of quantum mechanics, QM.

A classical analogy helps understand the subsequent time dependent linear Schrödinger equation of QM. Imagine a sea wall with two gaps, and a beach beyond. Let a series of plane waves approach the wall. As it passes through the gaps, each wave yields two semicircular wave patterns that approach the beach. If these semicircular patterns overlap at the beach, there will be points on the beach where the crests of the two wave patterns coincide, yielding a higher wave crest. Similarly there will be beach points where the troughs of two waves coincide yielding lower troughs. But there will also be points on the beach where the peak of one wave coincides with the trough of another wave and the two will cancel entirely.

The Schrödinger time dependent linear wave equation produces similar waves. Where peaks and peaks coincide, or troughs and troughs coincide, one obtains a bright bar of photons in “constructive interference”. Where peaks meet troughs, they cancel yielding dark bars in “destructive interference” and hence the interference pattern. An “action” variable in the equation keeps track of the phases in time and space of the Schrödinger waves.

Quantum “weirdness” arises due to the linearity of the equation, for sums and differences of solutions are also solutions. This linearity permits the famous Schrödinger Cat puzzle in which a cat in a box, prior to measurement, is *simultaneously* both dead and alive.

It is notable that, since 1927, no one knows what is “waving” in the Schrödinger wave equation. Meanwhile, von Neumann’s axiomatization of quantum mechanics (23), includes this propagating Schrödinger wave and the mysterious quantum measurement process. Here each wave has an amplitude. The *square* of the modulus of an amplitude, called the Born rule (24), yields the *probability* that that amplitude will be measured in von Neumann’s Process 1, or “R” process with its controversial “collapse of the wave function” of many amplitudes to only one, which can become classical as in the spot each photon makes on the screen of the two slit experiment. In general, there is, to the best of my knowledge, no agreed derivation of quantum measurement from within QM.

## Open Quantum Systems

The emergence of the classical world from the propagation of the Schrödinger wave equation is a deep mystery. One of the current best hypotheses requires distinguishing a quantum “system” from its “environment” yielding an “open quantum

system” and its “environment”. The key idea is that phase information within the open quantum “system” can be lost, acausally, to the quantum environment. This process is called “decoherence” (25). Then, within the system, the “action” gradually loses information about where the peaks and valleys of the Schrödinger wave “are”, so constructive and destructive interference cannot happen, nor can interference patterns. This interference hallmark of quantum effects is gradually lost and classicality is approached arbitrarily closely, reaching classicality “for all practical purposes”, FAPP.

Decoherence is well established experimentally. It disrupts quantum coherent qubit behavior in quantum computers.

Critically, decoherence is yielding new physics. First decoherence takes time. A typical time scale is a femtosecond. During that time phase information is being lost from the quantum system. The Schrödinger wave equation is time reversible. But decoherence is a dissipative process, so is not fully describable by the Schrödinger equation. New physics is expected and found.

I give three examples of this new physics. We are all familiar with the radioactive decay half life, due to closed quantum system Poisson distributed decay of the radioactive nucleus, whose integral is the familiar half life of exponential decay. In the confirmed Quantum Anti Zeno Effect, the decay is faster than any exponential (26). New physics.

Of interest to us as biologists, decoherence can alter the rate of chemical reactions (27). Decoherence happens in cells. What are the implications for molecular, cellular, neural, biomedical, drug and other behaviors? We don’t yet know.

An essential feature of decoherence is that the weird superposition states, the cat simultaneously dead and alive, decohere very rapidly, leaving what are called one or more “pure states”, if more than one, this is called a mixed state. Thus the cat is either dead or alive, but not simultaneously both. We don’t know which until quantum measurement (Seth Lloyd pc, Miles Blencowe, pc).

Recoherence, including to a new superposition state, is possible for open quantum systems. i. Several papers by Paz et. al., (28,29) and Briegel (30,31), show that a quantum entangled state can decohere to classicality FAPP and recohere again. ii. Imposition of a classical field can induce recoherence (32). iii. The Shor quantum error correction theorem (33), proves that if in a quantum computer some qubits are partially decoherent, measurement can be done and information injected, correcting the qubits back to full coherence.

In summary, and stunningly, for open quantum systems it is just becoming known that both decoherence to classicality FAPP and its *reverse*, recoherence, perhaps to a new quantum coherent superposition state, can occur.

Then, in principle, quantum degrees of freedom, including biomolecules, can “hover” between open quantum behavior and classicality FAPP. It is right to stress, as above, that this may have very large implications for the actions of molecules in cells, and drug discovery, design, and action. After all, we treat biomolecules as classical. We may be wrong.

## The Poised Realm

Gabor Vattay, a quantum physicist at Eotvos University Budapest, Samuli Niiranen, a Computer Scientist at the

Tampere University of Technology, Finland, and I, have proposed “The Poised Realm” between fully coherent quantum behavior in *open* quantum systems and classicality FAPP. Picture a two dimensional coordinate X, Y system. The Y axis rises from the origin, where there is open quantum coherent behavior, via decoherence, to classicality FAPP up the Y axis, and via recoherence down the Y axis to open quantum coherent behavior. The X axis is new, comprising “order”, “criticality” and “chaos”. The two axes box in the Poised Realm. The X axis, order, criticality, and chaos is well defined in the classical limit and now is being extended to embrace partially open quantum behavior in the presence of different extents of decoherence and recoherence.

Motion out the X axis from the origin, characterized classically by a frictionless pendulum, can be obtained in at least two ways. The first concerns the “Hamiltonian” of the classical system. A pendulum is perfectly ordered. If released from different initial heights, the frictionless pendulum describes roughly circular orbits in a coordinate space of position and velocity. These circular orbits are parallel, hence neither converge nor diverge. Mathematically, this lack of divergence or convergence is described by a 0 valued Lyapunov exponent. As one moves out the X axis, the Hamiltonian of the system changes. In the ordered regime, the Lyapunov exponent remains a constant 0. But when the Hamiltonian is deformed enough, at “criticality” the Lyapunov exponent becomes slightly positive, the onset of divergence of flows in state space constitute chaos. As the Hamiltonian is modified further, the Lyapunov exponent becomes more positive. This kink at “criticality” is a “second order phase transition”, and well established (34).

A second means to move out the X axis consists in using a “kicked quantum rotor”. A quantum rotor is a one dimensional hoop of states around which a quantized electric charge rotates. It can be kicked by a laser, with intensity K. As K increases in intensity, Vattay (pc), has shown that at first there are many amplitudes propagating, then few, then a single amplitude transforms to “classical” diffusive behavior in momentum space (35).

This classicality is reversible if K is decreased or the Hamiltonian is changed.

Thus, classicality, presumably FAPP, can be *reversibly* achieved up the Y axis or out the X axis.

## **The Non-Algorithmic, Non-Determinate, Yet Non-Random Trans -Turing System.**

I recall here the fully algorithmic Turing machine described above. Several points, sketched above, are essential. First, all contemporary computers are based on the Turing Machine. Second, the Turing machine is *completely definite*. It is the perfect instantiation, restricted to discrete space, time, and state, of classical physics and Descartes’ Res extensa machine world view. iii. This definite behavior of a Turing machine is the definition of algorithmic behavior. iv. Critically, a major contemporary view in neuroscience and computer science and much of the philosophy of mind is that the mind-brain system *must be* algorithmic - some huge system of interconnected logic gates or, more broadly, continuous time and state classical physics neurons firing.

I now describe non-algorithmic, non-determinate, but also non-random Trans-Turing systems. None has been constructed. I believe they are constructible. More the mind-brain system may be not only a vast *non*-algorithmic, non-determinate system, in contrast to classical physics in general, but also a non random Trans-Turing System. More broadly, classical physics is state determined. The mind brain system may be partially open quantum and Poised Realm, hence, via decoherence to classicality FAPP, or via quantum measurement, the mind-brain system may not be a state determined system.

The central ideas are simple. A Trans-Turing System, TTS, “lives in” the Poised Realm, and perhaps involves quantum measurement in the Poised Realm. i. There are quantum degrees of freedom propagating in short lived superposition states that decay rapidly due to decoherence. But these short lived superposition states undergo constructive and destructive interference and will be one basis for a *non*-Determinacy in the Trans-Turing system when coupled to decoherence to classicality for all practical purposes, FAPP, or quantum measurement. Thus TTS are not algorithmic, not determinate and not state determined, in contrast to a Turing machine.

Second, either via decoherence or motion out the X axis or both, quantum degrees of freedom become classical FAPP or via quantum measurement, become classical “Simpliciter”. Both decoherence and measurement are acausal and yield the non-determinant behavior of the Trans-Turing System.

Third, there are, in addition, coupled classical degrees of freedom in the TTS.

Fourth, when quantum degrees of freedom, and either superposition states or pure states become classical FAPP, or are measured, that *alters* in different specific ways the effects of the now classical degrees of freedom on one another, thus alters the non-random collective dynamics of the coupled classical degrees of freedom. In turn this altered non-random classical behavior alters non-randomly the behavior of remaining quantum degrees of freedom.

Fifth, in turn this non-random alteration of the behavior of the remaining quantum degrees of freedom alters non-randomly which of the open quantum degrees of freedom decohere or move out the X axis to classicality FAPP. In particular, higher quantum amplitudes tend to decohere with higher probability. So non-randomly altered quantum behavior, including altered constructive and destructive interference, alters non-randomly which amplitudes become higher, thus alters non-randomly which amplitudes decohere to classicality FAPP.

Sixth, in turn, classical FAPP degrees of freedom can recohere, for example, driven by a coherent electromagnetic field whose intensity and period distribution can be tuned non-randomly thereby injecting information. The recoherent degrees may achieve a new controlled superposition state, thereby altering non-randomly the constructive, destructive, and pure states behaviors among themselves and other quantum amplitudes, thereby non-randomly affecting which amplitudes achieve higher amplitudes and tend to decohere, and also non-randomly altering the behaviors of the coupled classical degrees of freedom in the TTS.

These six are the building blocks of a Trans-Turing System.

A part of a TTS has been realized in a computation by D. Salahub, a quantum chemist at U Calgary and colleagues, in JACS. Salahub et. al.,(36), considered a quantum system of many nuclei and many electrons. The system consists of two potential wells, say A and B. The vertical Y axis is energy. The X axis is a chemical reaction coordinate. The two potential wells overlap at some point in the X and Y plane, in what they call the “seam region”. Here at this seam the nuclei are in a superposition of states, simultaneously A and Not A, B and Not B. Via gradual decoherence the nuclei fall into one of the minima, either well A or well B, and become classical FAPP. But in turn this alters the effect of the now classical FAPP nuclei on the electron cloud which does not rapidly decohere. Thus, if the nuclei are now in well A the electrons behave differently than if the now classical nuclei FAPP are in well B and the electrons behave differently if the nuclei are still a superposition in the seam region.

This model is the first instantiation in quantum chemistry that I know in which some quantum degrees of freedom, here the nuclei in a superposition of A and Not A and simultaneously B and Not B, decohere to classicality FAPP, to well A or well B, and thereby alter the behavior of the remaining quantum degrees of freedom, the non-decohering electrons.

A more refined calculation would allow the many nuclei in this system to decohere in some sequential order. As they do, the newly classical FAPP nuclei will yield a sequential alteration in the behavior of at least the electrons and probably the remaining open quantum system superposition nuclei, as well as the other now classical FAPP nuclei. That research lies in the future as does study of such a system if the classical FAPP nuclei can be made to recohere to some perhaps new superposition state, perhaps by an external field, perhaps by interactions of many such subsystems within a molecule.

The essential points about Trans Turing Systems are:

- i. Their behavior is not Turing definite, both because of constructive and destructive interference of superposition states, followed by falling to a classical FAPP state where high amplitudes preferentially decohere, and remaining quantum pure states will also decohere probabilistically or by quantum measurement. Further, the total constructive and destructive interference behavior, and further controlled recoherence behavior, alter non-randomly which amplitudes achieve high amplitude so decohere preferentially to classicality FAPP with what probabilities, or are quantum measured, by the Born rule, with what probabilities. The ongoing behavior is not definite, hence NOT algorithmic. The behavior is not state determined. The behavior is also non-random.
- ii. The above behavior is, as noted, not “quantum random”, as in the case of radioactive decay, for a further reason: The classical degrees of freedom have their own Hamiltonian, hence non-random dynamics, which may, in addition, affect non-randomly the behaviors of the quantum degrees of freedom, hence which quantum amplitudes, via constructive and destructive interference, become high amplitudes and preferentially decohere and

when or are preferentially measured via the Born rule. The behavior is both non-deterministic, and non-random.

The TTS may receive quantum, open quantum, Poised Realm, and classical inputs and output open quantum, Poised Realm, and classical output behaviors. So it is a non-algorithmic, non-deterministic via decoherence to classicality FAPP or quantum measurement, yet non-random, information processing system. Consequently if TTS, as single or coupled systems are constructible, perhaps in liposomes, or nano-devices, we have a new non-algorithmic, not state determined, and not random “device”, unlike a Turing machine or logic gate, or deterministic classical physical system to consider for the mind-brain system. We no longer are almost forced to the conclusion that mind-brain *must be* classical physics, definite, and either discrete time and state logic gates or continuous time continuous variable deterministic “consciousness neurons, coupled into a huge network. TTS may also take us far beyond the Turing machine technologically.

## A Responsible Free Will

As noted above, the view that consciousness emerges from a vast network of classic physics logic gates or classical physics neurons may be entirely correct. However, it has a big price: We are deterministic so have no ontologically real responsible free will. Such a system could exhibit chaotic behavior, yielding the “illusion” of free will, but such a free will would not be ontologically real, for the classical physics neural system remains deterministic.

But there is another horn to this free will dilemma if we seek an ontologically real and responsible free will and then try to use standard quantum randomness. I have a radioactive nucleus in my brain, I walk down the street, the nucleus randomly decays, and I kill the little old man so my “free will” is ontologically real due to quantum indeterminism. But killing the old man is not my fault, just random quantum chance! I have no *responsible* free will if we use quantum randomness.

But a Trans-Turing system is both *not* deterministic, hence not algorithmic, and *not* quantum random, it is something entirely new. I hope this can break the horns of the standard responsible free will dilemma and allows for an ontologically real and responsible free will. I believe more is needed, building upon the idea of Ross Ashby’s famous homeostat (37), with its subset of “essential (classical physics) variables” that must be kept in bounds, to provide an internal “goal state” for the total system, to begin to yield a non-random but non-deterministic responsible free will.

This starting sketch, even if right, is inadequate. There is no mention of some analogue or actuality of sensory inputs, motor outputs or the capacity of a coupled Trans-Turing System, or set of entangled TTS systems, joined to the classical aspects of the brain, presumably classical physics neural networks, to classify its environments and act appropriately given goals and subgoals. Below, in proposing the testable hypothesis that qualia are associated with quantum measurement, it seems that “experiences” have as a natural dual, that which experiences, the rudiments of an

"I."Agency", on this view, is an elaboration of these rudiments in the total mind-brain system.

I note that R. Penrose (13,14), seeks a non-deterministic, non-algorithmic, yet non-random behavior of consciousness via a modified non-deterministic, so non-algorithmic, yet non-random quantum measurement process, "objective reduction", which might be associated with quantum gravity. Unlike Penrose, who may surely be right, I seek the same non-deterministic, so non-algorithmic, yet non-random behavior in Trans-Turing systems operating in the Poised Realm.

## Answering Descartes: How Can Mind Act On Brain

Due to the causal closure of classical physics, we have remained frozen with the Cartesian problem for 350 years. Mind has nothing to do and no way to do it. I believe that open quantum systems and the mind-brain system as one or trillions of interlocked Trans-Turing systems may afford an answer to Cartesian dualism, for it breaks the causal closure of classical physics. Decoherence is an *acausal process*. Thus if the mind brain system lies in the poised realm, decoherence of "mind" to classicality FAPP allows "mind" to have *acausal consequences for brain*, without acting causally on brain. We have indeed escaped the causal closure of classical physics.

But we want mind to do this many times in our lives. Trans-Turing systems, living in the Poised Realm, where recoherence, perhaps to new superposition states, allows mind to repeatedly decohere to have acausal consequences for material brain.

Quantum measurement can occur in Trans-Turing systems. But quantum measurement, von Neumann's Process 1 or "R" process, is also acausal, and also allows mind to have acausal consequences for brain. More, even should von Neumann's Process 1 or "R" depend upon the Born rule and his square of the amplitudes to achieve the probability of its acausal measurement, the ongoing behavior of the Trans-Turing systems modifies *non-randomly* which amplitudes are propagating and which achieve high amplitudes and tend to decohere or be measured, so the total behavior is non-random. Once measured a classical degree of freedom can flower again into quantum behavior again, allowing repeated acausal mind-brain action. The non-random but non-determinant total behavior may support a responsible free will.

## Res Potentia and Res Extensia Linked by Quantum Measurement

I now come to the most radical proposition in this chapter. It can be false and the remainder of this chapter stay largely intact. I am, with proper hesitation, about to propose a new dualism, Res potentia, the realm of the ontologically real Possible, and Res extensa, the realm of the ontologically real Actual, linked - hence united - by quantum measurement. The very basis of this is quantum mechanics itself.

I turn first to the late 19th Century American philosopher C.S. Pierce (3). He noted that Actuals and Probables obey Aristotle's Law of the Excluded Middle. Here it is: The table is or is not in the room. There is nothing "in the middle".

Hence the statement, "The table simultaneously is and is NOT in the room" is a contradiction, always false. Now consider: "The probability of 5234 heads out of 10,000 fair coin flips is simultaneously 0.245 and is not 0.245". It too is a contradiction, always false. Classical physics obeys Aristotle's Law of the Excluded Middle. But, said Pierce, "Possibles" evade the law of the Excluded Middle. "A is possibly true and A is possibly not true." is NOT a contradiction.

Now consider Richard Feynman's (2), "sum over all possible histories" formulation of quantum mechanics, agreed by all to be an equivalent formulation."A photon on its way through the two slits, simultaneously takes all possible pathways through the two slits to the photoreceptor." It follows that the single photon "simultaneously possibly *does* and possibility *does not* pass through the left slit". This is *not* a contradiction.

The critical implication is that Feynman's formulation of quantum mechanics *evades* Aristotle's Law of the Excluded Middle. Therefore, I claim, Feynman's formulation of quantum mechanics is fully interpretable in terms of ontological *real* Possibles, Res potentia. The unmeasured Schrodinger wave concerns Res potentia. Res potentia proposes an answer to what the unmeasured Schrodinger wave is "about".

This is a huge step, not to be taken lightly. I note that Aristotle himself toyed with the reality of "potentia". And British philosopher Alfred North Whitehead in Process and Reality, 1929,(4), proposed ontologically real Possibles which gave rise to ontologically real Actuals which gave rise to ontologically real Possibles.  $P \rightarrow A \rightarrow P \rightarrow A$ .

The idea may be radical, and may be right, but I am not the first to propose it. We will find evidence consistent with the reality of Res potentia below in the Conway Kochen Strong Free Will Theorem. Further, outstanding quantum physicists are very close to the concept of Res potentia. I quote Dieter Zeh:

"in classical physics you can and do assume that only one of the possibilities is real (that is why you call them possibilities). It is your knowledge that was incomplete before the observation. Mere possibilities cannot interfere with one another to give effects in reality. In particular, if you would use the dynamical laws to trace back in time the improved information about the real state, you would also get improved knowledge about the past. This is different in quantum theory (for pure states): In order to obtain the correct state in the past (that may have been recorded in a previous measurement), you need all apparent "possibilities" (all components of the wave function - including the non-observed ones). So they must have equally been real." (38).

Clearly Zeh is saying, "possibilities"... "must have been equally real." Res potentia removes the quotes from "possibilities" to propose an ontologically real Res potentia.

More, a founder of quantum mechanics, W. Heisenberg, often spoke of "Potentia" sometimes as "Probabilities" (39), sometimes as "Possibilities" (40), as a separate ontologically real realm along with ontologically real Actuals. I am following Heisenberg with my Res potentia as a realm of ontologically real Possibles.

See M. Epperson (41), for a cogent discussion of many quantum authors and an ontological dualism based on real

Actuals and real “Probables” which DO obey the Law of the Excluded Middle. I stress again that unlike Descartes Res cogitans and Res extensa, never united, Res potentia and Res extensa truly *are* united by quantum measurement

## What is Consciousness

Philosopher of mind Jerry Fodor quipped that “Not only have we no idea what consciousness “is”, we have no idea what it would be like to have an idea what consciousness “is” (42).

To my surprise, Res potentia leads to an obvious idea about what consciousness IS. Consciousness is a participation in The Possible, an ontologically real Res potentia.

I offer three pieces of evidence:

1. Where is the possibility I will skate across town reading the NY times and not be hit by a car? I think we all feel that the “possibility” itself is not spatially locatable, it is not spatially extended.
2. Now consider your experienced visual field. Where is your experienced field located? I think we all sense that our experienced field is not located spatially. It is not spatially extended.
3. Just below I will propose that qualia are associated with quantum measurement and further below hypothesize that entanglement of many quantum degrees of freedom, perhaps among neurotransmitter receptor molecules in anatomically unconnected synapses in the brain, may, by each being quantum measured, yield causally non-local Einstein, Podolsky, Rosen, EPR, high correlations of now bound qualia (20), to solve the “qualia binding problem” in neurobiology. Non-local correlations are “non local” because they are beyond speed of light signaling and “instantaneous”, hence also “non-spatial”.

This non-spatial character of “Possibilities”, Experience and Non-Local EPR quantum measurements may be happenstance or a clue. Taking this parallel as a clue may lead us forward in new ways.

## Qualia may be Associated With Quantum Measurement

Where is it natural to locate experience itself, the blueness of blue, the taste of wine, qualia? I suggest qualia are associated with quantum measurement, ie Possible “becomes” Actual, Possible  $\rightarrow$  Actual. As we shall see, this leads to testable consequences. It is not a bald hypothesis standing alone, for as just noted I will propose below that quantum entanglement among many quantum degrees of freedom in anatomically unconnected synapses, and non-local EPR correlations achieved by a set of quantum measurements of these entangled degrees of freedom may help solve the “qualia binding problem” and the Unity of Consciousness issue. Thus solving the binding problem may require the hypothesis that quantum measurement is associated with qualia. The hypothesis should be testable in the brain. More, entanglement to solve the binding problem is testable. I note that physicist

H. Stapp has different but somewhat related ideas (43) See also Penrose (14).

A critical feature of quantum measurement, my physicist friends assure me, is that it has never been derived from within quantum mechanics. Granted Res potentia, such a derivation may be disallowed. “X is Possible” does *not* imply “X is Actual”. Our difficulties with such a derivation since 1927 may be ontological, not technical - mathematical. If Res potentia is ontologically real, the same ontological issue may bear on our failure to unite General Relativity and Quantum Mechanics: the “X is Possible” of unmeasured quantum mechanics does *not* imply the “X is Actual” of General Relativity.

On Res potentia, a second feature of measurement becomes equally important. What is the “becomes” of Possible “becomes” Actual? What is the status of “ $\rightarrow$ ” in  $P \rightarrow A$ ? It is not a classical becoming like water freezing, nor the unitary propagation of the Schrodinger wave. As a “becoming” it seem not to be an *existing state* at all. Qualia are a “becoming” not an “existence”. Nor can “ $\rightarrow$ ” be a mathematizable deductive entailing process, for if it were, it would enable deduction from “X is Possible” to “X is Actual”, which is invalid if Res potentia is real. Then there is no mechanism for the quantum measurement captured in von Neumann’s “ad hoc” Process 1 or “R” projection process.

The above paragraph depends upon the reality of Res potentia. But the proposal of a real Res potentia ties to the recent, 2009, Conway Kochen Strong Free Will Theorem (44), which states that if physicists have free will so do electrons, that the world is non-determinant, that there can be no mechanism for quantum measurement, and that the relevant property does not exist before measurement. This theorem rests on free will for the physicist. But above I have argued that Trans-Turing systems in the Poised Realm, *without relying on an ontologically real res potentia*, may afford an ontologically responsible free will. Responsible free will may well require qualia, experience, which I propose is associated with quantum measurement. This again is a proposal that does not require the reality of Res potentia. But a responsible free will supports the claims of the Strong Free Will Theorem. Conversely, this theorem states that, given the free willed physicist, the world is non-determinant. This is consistent with the hypothesis of the reality of *res potentia*. More, by this theorem, if qualia are associated with quantum measurement, there is no mechanism for that measurement. But measurements yield classical degrees of freedom that, as such, can have classical causal effects on the classical world. Mind, qualia, can, via acausal measurement, act causally on the world classically. Perhaps, as I propose below, neurotransmitter receptors are the loci of quantum measurement and qualia. Then the classical variable consequences could alter post synaptic voltage gate behaviors leading to neural firing or not. In turn qualia themselves emerge as irreducible.

The vice of this view is that it hides the mystery of qualia in the mystery of measurement. The virtue of this “hiding” is that it may explain, at last, why we cannot isolate or pin down an irreducible character of qualia. Philosopher David Chalmers (45), also proposes on independent grounds that qualia are irreducible.

I stress that this hypothesis does not say what qualia *are*.



This hypothesis is testable. Anesthetics bind to hydrophobic pockets in neurotransmitter receptors in synapses (46). If they freeze receptors so they cannot quantum measure, no more qualia can arise. Moreover, *Drosophila* can be anesthetized by ether. Select for ease of being anesthetized and seek the molecular components involved in easy anesthetization. The normal, or wild type, versions of these molecules may be involved in consciousness and their quantum and quantum measurement properties studied.

If I assume qualia are associated only with measurement, a potential role for unmeasured quantum behavior in the mind-brain system could be unconscious mental processing, which may have classical consequences via decoherence to classicality FAPP, without measurement. Possibly this bears on Libet's results of neural activities 200 or more milliseconds before conscious awareness of a decision to act (47). This too should be testable.

If qualia ARE associated with quantum measurement, it seems natural that the dual of "experience" is that which experiences, a rudimentary "I". From this rudimentary "I" in the entire mind-brain system with its inputs and outputs, my hope is that full "Agency" and an ontologically real and responsible free will can be found.

## Standing the Brain on its Head

I begin with a stunning fact. The Box jelly fish, with only a loose neural net, no evolved brain, but eyes that have evolved to see shape and color, swims at five knots adroitly avoiding obstacles (48). An evolved brain is not needed for these feats. Also choanoflagellates, single cell precursors to the animals, have many molecular components of synapses (49).

Many readers of the chapter know the neuroanatomy of the human brain and much of its physiology. In brief, we have about 10 to the 11th neurons, each with an average of 6000 synapses. Cell bodies have descending axons which may or may not branch, but each ends on synapses associated by synaptic spikes on dendrites in arborizations which lead into cell bodies. When an action depolarization potential travels down an axon to a synapse, presynaptic vesicles release one of a set of neurotransmitters, such as GABA, which crosses the synaptic cleft to the adjacent dendrite of the post synaptic neuron, and binds to post synaptic neurotransmitter receptors which are often in clusters of many proteins. In turn, often this leads to opening of an ion channel, a transient flow of ions, and a very short term depolarization or hyper-polarization (excitatory or inhibitory respectively) of the tiny local patch of dendritic transmembrane potential. These local changes flow to the cell body and are summed. If the resulting transmembrane potential at the cell body is more than - 20 mV, an action potential is initiated and travels down the axon. Most neurobiologists think classical physics action potentials in neurons carry a "neuronal code" underlying consciousness.

In Francis Crick's Astonishing Hypothesis (1), he notes in a throw away line, that vast amounts of information about tiny time-space alterations in dendritic transmembrane potentials and behaviors of synaptic molecules are thrown away in neural classical physics action potentials.

What if we consider "standing the brain on its head", and supposing that this vast amount of information in and around synapses and local dendritic regions are the "business end" of

the brain- sensory-motor system. This does not vitiate at all the huge amount of work on neural circuitry and classical action potentials and information processing by classical neural action in the brain.

However, it does raise the possibility that the "neural correlates of consciousness" may lie in synaptic and local dendritic, possibly poised realm behavior, possibly in quantum measurement processes.

I note that Beck and Eccles have considered quantum processes in synapses (50)

## Quantum Entanglement, Niiran's idea, and the Binding Problem

Consider, says Crick, a yellow triangle and blue square. Let "yellow", "triangle", "blue", and "square" be processed in different, anatomically unconnected brain areas. How are they bound into yellow triangle and blue square. This is the binding problem. Crick focuses hope on squeezing perhaps millions of distinct sets of features to be bound into different phases of the 40 Hertz oscillation, as I have described.

The first idea I propose is to use quantum entanglement to link quantum processes in different, anatomically unconnected synapses to start to solve the binding problem. Entanglement occurs if a quantum degree of freedom, say a photon, decays into two lower energy photons that go off in opposite directions, even so far apart that even light cannot travel between them in the *interval between quantum measurements of the two entangled photons*. QM says, and it is confirmed over and over, that the *two quantum measurements* will be highly correlated, even though no light or information can have traveled between the two sites. This is "EPR non-local correlation" (20). I stress that in the entangled state, the two photons remain in a *single* quantum state.

I want to try to use quantum entanglement among many synaptic degrees of freedom to try to solve the Binding Problem. Hence, as I have emphasized, it is very attractive to me that these quantum correlations require quantum *measurement* of the entangled degrees of freedom, and I have already supposed that quantum measurement itself is associated with qualia. Then these many entangled degrees of freedom in a *single* quantum state when measured yield qualia that *are* bound. The hypothesis that qualia are associated with quantum measurement does not stand alone, it may afford a part of an answer to the Unity of Consciousness.

Clearly, such entanglement may require long range entanglement among anatomically unconnected synapses and neurons in the brain connecting the "right" set of, say, synaptic molecules. How and whether this may be accomplished is, at present, uncertain, but see below.

Samuli Niiranen had a lovely idea. "If you measure the position and momentum a single classical gas particle in a box, do you know about the shape of the box?" No you do not. "But", he continued, "a quantum wave process in a potential well that serves as its boundary condition "knows" about the shape of that potential well, for example in its measured energy spectrum!". He is right.

Think of music in a room and trying to describe air pressure waves using bits. Now think of 1000 differently shaped drum heads well placed in the room. Their patterns of

vibrations, ie the eigenfunction spectra of the drums bound to drumheads, “know” the music in the room, and do so in an analog embodied way, not a digital or propositional way. A telephone is not digital either.

This leads to the idea that the brain’s sensory system and the whole brain, may tune the synaptic or local dendritic transmembrane potentials in tiny time-space regions of the brain, such as synapses and adjacent local dendritic membranes so they jointly “cover”, like many tuned antennas, the visual scene such that quantum processes in those potential wells, when entangled and measured yield both a Unity of Consciousness and solve the Binding Problem in an analog not digital way.

Two final points. It now appears that increasing the number of entangled degrees of freedom *increases* the quantum EPR correlations. This increase is the opposite of the curse of dimensionality. This helps the binding problem. Second, local actions can alter which quantum degrees of freedom are entangled, perhaps offering an account for serially shifting focus of attention, and might entangle the “correct” set of quantum degrees of freedom for each focus of attention (51).

Can all this be correct? I certainly do not know. But the ideas seem coherent, testable, and jointly seem to offer new purchase on manifold problems.

### “Programming Trans-Turing Systems”

We have known about Turing Machines since the mid 1930s and programming the von Neumann architecture for over fifty years. We have no experience with Trans-Turing Systems, TTS. But we face a problem: How would we achieve a TTS that “does something we want”?

There seem to be two approaches. Simulate the TTS on a digital computer and evolve a population of TTS, or a population of interacting entangled, measured, TTS, to yield the behavior desired. This is analogous to the Genetic Algorithm of Holland (5).

Another approach which may be worth considering is creating self reproducing molecular systems, perhaps autocatalytic sets of polymers in dividing liposomes, supplied with energy by pyrophosphate or in other ways, and capable of open ended evolution. Recent work shows that: 1) collectively autocatalytic sets arise as the diversity of polymers in a reaction set is increased (52,53). 2) Such systems can undergo open ended evolution (54). 3) Liposomes can grow and divide (55). 4) A collectively autocatalytic set in a reproducing container can yield synchronization of the reproduction of each (56). Experimental collectively autocatalytic sets have been constructed (57). Libraries of stochastic DNA, RNA, peptides, polypeptides and proteins can be made (58), so we can test for the emergence of collectively autocatalytic sets.

It is an exciting prospect that work on the origin of life and work on Trans-Turing Systems may come together. More Darwinian preadaptations among such co-evolving protocells generate new, unprestatable biological functions that maintain one or more such protocells, hence solve the frame problem (59). Co-evolving TTS in protocells may well solve the frame problem too.

Work with minimal cells as vehicles for TTS evolution and co-evolution may be possible (60).

In addition, nanotechnology, perhaps with populations of nano-devices that can be subjected to Holland’s Genetic algorithm (5), may prove useful.

## Conclusions

I have argued that classical physics Turing machines as models of the mind are possible, but leave us at best with no free will, and an epiphenomenal consciousness. I believe that we can begin to go beyond Turing, to create non-algorithmic, non-determinate, and non- randomly behaving Trans-Turing systems, living in the Poised Realm, perhaps in self reproducing protocells, perhaps as nano-devices, both open to evolution or co-evolution to achieve useful ends. I propose tentative answers to Descartes about mind and body. Many of the ideas in the Chapter are new science or even radical. They may portend transformations in quantum physics, quantum chemistry, a new Poised Realm behavior of biomolecules hovering between quantum and classical behaviors, a new approach to neurobiology, the philosophy of mind, and the radical possibility of Res potentia with consciousness a participation in The Possible, qualia as irreducible and associated with quantum measurement which also may be irreducible, and entanglement and quantum measurement to achieve a unity of consciousness. I hope these concepts point the way forward for us all.

## Acknowledgements

The author was partially supported as a Finnish Distinguished Professor at the Tampere University of Technology by the TEKES Foundation.

## References

1. Francis Crick, The Astonishing Hypothesis: The Scientific Search For the Soul, TOUCHSTONE, Simon an Schuster, N.Y. 1994.
2. Feynman, R. P. (1948). The Space-Time Formulation of Nonrelativistic Quantum Mechanics. Reviews of Modern Physics, 20: 367-387.
3. C.S. Pierce Collected Papers Vols I and II, Edited by Charles Hartshorne and Paul Weiss, Harvard University Press, Cambridge Mass.1960.
4. Alfred North Whitehead. Process and Reality: An Essay in Cosmology, Corrected Edition. Ed. D. Griffin and D. Sherburne. Free Press. N.Y. 1978.
5. Holland John, H. Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Mich. 1975.
6. Alan Turing, Can Automatic Calculating Machines be Said to Think?, in Copeland, B. Jack, The Essential turing: The ideas that gave birth to the computer age, Oxford University Press, Oxford, 1952.
7. Bertrand Russell, 1912, The Problems of Philosophy, Oxford University Press, 1997 N.Y.
8. Ludwig Wittgenstein (1921) Logisch-Philosophische Abhandlung, Wilhelm Ostwald (ed.) Annalen der Naturephilosophie, 14.
9. Ludwig Wittgenstein, Philosophical Investigations, Blackwell Publishing Oxford, 1953/2001.
10. McCulloch, W. and Pitts, W. (1943) A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical biophysics, 7: 115-133.

11. Paul J. Werbos. *The Roots of Backpropagation. From Ordered Derivatives to Neural Networks and Political Forecasting.* N.Y. John Wiley and Sons, Inc, 1994.
12. J.J. Hopfield. (1982). Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Science of the USA*, vol 79, no 8 pp 2554-2558.
13. Roger Penrose, *The Emperor's New Mind: Concerning Computers, Minds and the Laws of Physics*, Oxford University Press, Oxford. 1989.
14. Roger Penrose, *Shadows of the Mind: A search for the missing science of consciousness.* Oxford University Press, Oxford, 1994
15. Daniel C. Dennett. *Consciousness Explained.* Little Brown. 1991.
16. John R. Searle. *The Mystery of Consciousness.* Th New York Review Books, N.Y. 1997.
17. Gerald Edelman. *Bright Air, Brilliant Fire: On the matter of mind.* Basic Books, N.Y. 1992.
18. Rene Descartes, *Discourse on Method.* 1637A.D. Open Court Publishing, Chicago, 1899, Reprint 1962.
19. Schrodinger, E. (1926). An Undulatory Theory of the Mechanics of Atoms and Molecules. *Physical Review* 28,(6) 1049-1070.
20. A. Aspect, Dalibard, J., and Rogers, G. (1982). Experimental Test of Bell's Inequalities Using Time-Varying Analyses. *Phys. Rev. Lett.* 44: 1804-1807.
21. A. Ishizaki, Calhoun, T.R. Schlou-Cophen, G.S. , Fleming, G.R. (2010). Quantum coherence and it s interplay with protein environments in photosynthetic electronic energy transfer. *Phys Chem Chem Phys.* 12 (27): 7319-7337.
22. R. P. Feynman., Leighton, R., and Sands, M. *The Feynman Lectures on Physics*, Vol 3: 1-8. Addison-Wesley. N.Y.1964.
23. John von Neumann. (1932). *Mathematical Foundations of Quantum Mechanics.* Beyer, R.T. trans. Princeton University Press. Princeton N.J. 1996 edition.
24. Max Born. *Atomic Physics*, 8th Ed. Dover, N.Y. 1989.
25. W. H. Zurek. (2002). Decoherence and the Transition from Quantum to Classical - Revisited. *Los Alamos Science Number* 27: (2002).
26. M.C Fischer, B. Butierrez-Medina, and M.G. Raizen. (2001). Observation of the Quantum Zeno and Anti-Zeno Effects in an Unstable System. *Phys. Rev. Lett.* 87(4) pp 040402-1- 040402-4.
27. O. V. Prezhdo. (2000). Quantum Anti-Zeno Acceleration of a Chemical Reaction. *Phys Rev Lett* 85: 4413-4417.
28. J. P. Paz, and Roncaglia. A. J. (2009). Entanglement dynamics during decoherence. *Quantum Inf Process* 8: 535-548.
29. C. Cormack and Paz, J. P. (2010). Observing different phases for the dynamics of entanglement in an ion trap. *Phys. Rev.A.* 81: 022306.
30. J. Cai, Popescu, S., Briegel. H. J. (2008). Dynamic Entanglement in Oscillating Molecules. *arXiv:0809.4906v1 [quant-ph]* 29 Sept 2008.
31. H.J.Briegel and Popescu. S. (2008). Entanglement and Intra-Molecular Cooling In Biological Systems - A Quantum Thermodynamic Perspective. *arXib:0806.4552v1 [quant-ph]* 27 June 2008.
32. C. Mainos. (2000). Laser induced coherence in ultraphoton excitation of individual molecules. *Phys. Rev. A* 61: 063410-6.
33. W. P. Shor. (1995). Scheme for reducing decoherence in quantum computer memory. *AT&Bell Laboratories.*
34. R.S. Mackay. *Renormalization in area preserving maps.* World Scientific N.Y. 1993.
35. G. Vattay (2010). Transition from quantum to classical behavior in kicked quantum rotors as kicking intensity increases. *Personal communication.*
36. A. de la Lande, Rezac, Levy, J.B., Sanders, B.C., Salahub, D. R. (2011). Transmission coefficients for chemical reactions with multiple states: the role of quantum decoherence. *J. Am. Chem. Soc.*13: 3883-3894.
37. R. Ashby *Design for a Brain: The origin of adaptive behavior.* Chapman & Hall. N.Y.1952.
38. D. Zeh. <http://www.fqxi.org/community/forum/topic/39> (3rd comment from top).
39. Werner Heisenberg. "The Development of the Interpretation of the Quantum Theory, in Neils Bohr and the Development of Physics, ed. Wolfgang Pauli (McGraw-Hill) N.Y. 1955.
40. Werner Heisenberg. *Physics and Philosophy: The Revolution in Modern Science.* Harper and Row, N.Y. 1958.
41. Michael Epperson. *Quantum Mechanics and the philosophy of Alfred North Whitehead.* Fordham University Press, N.Y. 2004.
42. Jerry Fodor. *The Mind Doesn't Work That Way: The Scope and Limits of Computational Psychology.* MIT Press. 2000.
43. Henry P. Stapp. *Mindful Universe: Quantum Mechanics and the Participating Observer.* Springer, 2007.
44. J. H. Conway and S. Kochen (2009). The Strong Free Will Theorem. *Notices of the AMS* 56 (2): 226-232.
45. David J. Chalmers. *The Conscious Mind: In Search of a Fundamental Theory.* Oxford University Press. N.Y.1996.
46. S. R. Hameroff. (2006). The Entwined Mysteries of Anesthesia and Consciousness: Is there a common underlying mechanism? *Anesthesiology*, 105: 400-412.
47. Libet, B. Cerebral processes that distinguish conscious experience from unconscious mental functions. In J. C. Eccles and O.D. Creutzfeldt, editors, *The Principles of Design and Operation of the Brain*, pp 185-205, Springer, 1990.
48. S.C. Morris in *Atoms & Eden: Conversations on Religion and Science*, Steve Paulson. Oxford University Press. N.Y. 2010.
49. King,N., Hittinger, C.T., Carroll S. B.. (2003). Evolution of key cell signalling and adhesion proteins predates animal origins. *Science* 301 (5631):361-363.
50. F. Beck and Eccles, J. C. (1994). Quantum processes in the brain" A scientific basis for consciousness. In Naoyuki Osaka (Ed.), *Neural Basis of Consciousness.* Amsterdam, Philadelphia: John Benjamins.
51. *Science News* Nov 20, 2010. Inducing Entanglement.
52. S. A. Kauffman. (1986). Autocatalytic Sets of Proteins. *J. Theor. Bio.* 119: 1-24.
53. W. Hordijk, Hein, J. and Steel, M. (2010). Autocatalytic sets and the origin of life. *Entropy* 12 (7): 1733-1742.
54. C. Fernando, Vasas, V., Santos, M., Kauffman, S., and Szathmary, E. (2011). Spontaneous Formation and Evolution of Autocatalytic Sets within Compartments. Submitted.
55. P.L. Luisi, Stano, P., Rasi, S., and Mavelli, F. (2004). A possible route to prebiotic vesicle reproduction, *Artificial Life*, 10: 297-308.
56. A. Filsetti, Serra, R., Carletti, T., Villiani, M.,and Poli., I. (2010). Non-linear protocell models: synchronization and chaos. *Eur. J. Phys. J. B* 77: 249-256.
57. N. Wagner and Ashkenasy, G. (2009). Symmetry and order in systems chemistry. *The Journal of Chemical Physics*, 130: 164907-164911.
58. Stuart Kauffman. *Origins of Order: self organization and selection in evolution.* Oxford University Press. N.Y. 1993.
59. Stuart Kauffman. *Reinventing the Sacred*, Basic Books, NY. 2008
60. R.L.Hotz. (2010). Scientists Create Synthetic Organism. *Wall Street Journal* May 21

# On the Growth of Brain Tumours: enhancing imaging techniques, highlighting limitations of current imaging, quantifying therapy efficacy and estimating patient life expectancy

James D. Murray<sup>1,2,3</sup>

<sup>1</sup> University of Oxford, UK

<sup>2</sup> Princeton University, USA

<sup>3</sup> University of Washington, USA  
james.murray@maths.ox.ac.uk

## Abstract

The prognosis for patients with high grade brain tumours (gliomas) is grim and the various treatment protocols such as surgery, radiation and chemotherapy cannot effect a cure. I describe, without any technical details, a simple but very practical model which uses patient data and brain scans to quantify the spatio-temporal growth of such brain tumours. Analysis of the model shows how difficult it is to decide on the tumour volume to be treated and shows why such treatments have so little success. The model simulations can estimate life expectancy for individual patients and show how it is possible to use the patient's past record to predict the efficacy of possible treatments. Recent patient data indicates that calculating such an index of treatment efficacy is indeed a realistic aim. With the increasing discussion about cell phone use and a possible increase in brain tumours, I describe how to obtain an estimate for when a brain tumour started given its size at detection.

## Introduction

High grade brain tumours, glioblastoma multiforme (GBM), are the most aggressive brain tumours and make up more than 50% of all brain tumours. There is 100% mortality rate for patients with such tumours with an approximate median life expectancy of 9-12 months. The various treatment protocols such as surgery (resection), radiation and chemotherapy cannot effect a cure but can sometimes extend survival time. Treatment efficacy depends on various factors such as where the tumour is located in the brain and the size of the key parameters, namely the growth rate and the diffusion rate. Diffusion in white matter is larger than in grey matter. It is the aggressive infiltration of cancer cells which make treatment protocols so difficult to localize. In spite of increasing accuracy of imaging techniques they still cannot detect cancer cell densities sufficiently accurately. The inadequacy of medical imaging is substantiated by the fact that irrespective of the extent of surgical resection or focused irradiation of the tumour it is always followed by multifocal tumour recurrence at or near the edge of the resected volume (Silbergeld *et al.* 1991).

A basic practical model which encompasses the two key elements in the growth of such tumours, namely the invasive

diffusive properties of the cancer cells and their growth rate is qualitatively given by the equation:

$$\begin{aligned} \text{Rate of change of tumour cell density} \\ = \text{diffusion (invasion) of tumour cells} \\ + \text{net proliferation of tumour cells} \end{aligned} \quad (1)$$

The mathematical form which quantifies the various terms in (1) is

$$\frac{\partial c}{\partial t} = \nabla \cdot D(x) \nabla c + \rho c \quad (2)$$

where the various terms in this equation are defined as:

- $c(x,t)$  = glioma cell density, cells/mm<sup>3</sup>, which is a function of the position,  $x$ , in the brain at time.
- $t$  = time, measured in months.
- $D(x)$  = diffusion (invasion), mm<sup>2</sup>/month, which quantifies the invasiveness of the cancer cells at position  $x$  in the brain.
- $\rho$  = proliferation rate (/month) of the cancer cells which gives the turnover time as  $\log 2/\rho$  (months).

The solutions of (2) are unbounded as time increases because of the form of the growth term which implies exponential growth. A more accurate model has in place of  $\rho c$  the expression  $\rho c(1-c/k)$  where  $k$  is a constant associated with the maximum concentration possible in the brain tissue. This equation, with a constant diffusion coefficient, is a classical population equation known as the Fisher-Kolmogoroff equation (Murray 2002). Solutions of it are bounded and exhibit traveling waves. However, in the time scales relevant to glioma growth and patient survival time it does not contribute significantly to the solutions relevant to cancer patients.

With two individual patient brain scans, such as CT, MRI and others, the key model parameters, namely diffusion and cell growth can be calculated. With these we can then predict the subsequent growth of such brain tumours. As illustrated below, analysis of the model shows how difficult it is to decide on the tumour volume to be treated and shows why such treatments can have little success. The model simulations can estimate life expectancy for individual patients and how to

predict the efficacy of different treatments. Patient data indicates that calculating such an index of treatment efficacy is a realistic aim. With the increasing debate on the possible increase in brain tumours as a result of cell phone radiation, realistic and scientific clinical trials will require information such as when a tumour started, how fast it grows and where it is in the brain outside of what can be detected with current brain imaging techniques. Here we show how the model provides a means of estimating the time from tumour initiation and life expectancy from tumour detection for individual patients.

The original model was first proposed and analysed in various situations. The brain was considered to be homogeneous matter bounded by the ventricles and skull (Cruywagen *et al.* 1995). Even with such a simple anatomical model the predictions of the analysis were broadly in line with patient observation of both low and high grade brain tumours. The limitations of current imaging techniques were clear. The model was then used to mimic various accepted medical treatments, specifically radiation, surgical resection (Woodward *et al.* 1996) and chemotherapy (Tracqui *et al.* 1995), Swanson *et al.* (2002), Rockne *et al.* (2010). A three dimensional model was proposed and studied by Burgess *et al.* (1997) who were the first to demonstrate that cancer cell diffusion, mainly ignored up to that time, is a major component of glioma growth. They showed that only those tumours with a low diffusion rate could benefit from wide surgical resection although eventually there will be multifocal recurrence. See Murray (2003) for a full discussion and review which encompasses anatomically correct brains.

### Virtual gliomas: enhanced imaging and current limitations

A major advance in the practical application of the model (1) was the availability of the brain web atlas (Collins *et al.* 1998). This allowed the model to be applied to anatomically correct brains (Swanson *et al.* 2002, 2004, Murray 2003). Among other things it made it possible to refine the gross anatomic boundaries and to vary the degree of motility of glioma cells in grey or white matter: these are biologically significant.

With the BrainWeb it was possible to solve equation (2) in a three dimensional anatomically correct brain in which the grey and white matter is clearly delineated.

The procedure is to evaluate the tumour size from brain scans and, crucially, estimate the parameter values for each patient to obtain the average diffusion coefficient and the average growth rate. There is a lower threshold of detection of cancer cells with all imaging techniques, whether CT or MRI, such as T1Gd and T2 imaging, or microscopic studies. To use the predictive potential of the mathematical model (2), serial imaging of the tumour was used to calculate its volume which was then taken as the volume of an equivalent sphere with radius  $r$ , namely  $4\pi r^3/3$ . We then consider the model to be radially symmetric with a constant diffusion coefficient, based on averaging the values from imaging. Equation (2) then becomes

$$\frac{\partial c}{\partial t} = D \left[ \frac{\partial^2 c}{\partial r^2} + \frac{2}{r} \frac{\partial c}{\partial r} \right] + \rho c \quad (3)$$

We consider that at time  $t=0$  there is a concentrated number of cancer cells,  $N$  cells/mm<sup>3</sup>, at  $r=0$  in which case the solution of (3) is given by

$$c(r, t) = \frac{N \exp(\rho t - \frac{r^2}{4Dt})}{8(\pi Dt)^{3/2}} \quad (4)$$

If the smallest level of image detection is denoted by  $c_1$  cells/mm<sup>3</sup>, then the radius,  $r$ , of the tumour for this cell density is, from (4) on solving for  $r$ ,

$$r = 2t\sqrt{D\rho} \sqrt{1 - \frac{1}{\rho t} \log\left(\frac{c_1}{N}(4\pi Dt)^{3/2}\right)} \quad (5)$$

For large time,  $t$ , the solution (5) gives the radius of detectable tumour and the velocity of growth,  $v$ , as approximately

$$r = 2t\sqrt{D\rho} \Rightarrow v = r/t = 2\sqrt{D\rho} \quad (6)$$

That is, the equivalent radial growth is linear in time.

### Approximate *in vivo* patient survival time

If we consider detection is when the spherical equivalent tumour volume is of radius 15mm and that death occurs when the radius is 30mm the approximate survival time from detection, in the absence of any treatments, is given, from (6), by

$$\begin{aligned} \text{Survival time (months)} \\ = t_{\text{survival}} = t_{r=30} - t_{r=15} = \frac{7.5}{\sqrt{D\rho}} \end{aligned} \quad (7)$$

Typical growth rates vary quite widely, approximately from 1-5 /month and diffusion rates from 1-8 mm<sup>2</sup>/month. The medians for 9 patients in the study by Rockne *et al.* (2010) are  $D=0.9$  mm<sup>2</sup>/month and  $\rho=1.16$ /month which gives a median survival time of 7.34 months.

Survival time, however, depends on where the tumour is mainly situated. If it is primarily in the grey area of the thalamus, for example, the diffusion is smaller and so the survival time is longer, as is clear from (7).

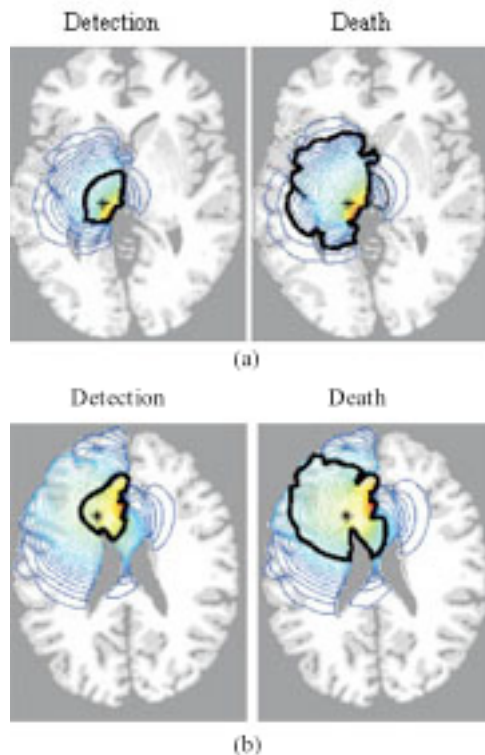
The diffusion in grey matter,  $D_g$ , is smaller than that in white matter,  $D_w$ : they can vary by as much as 100-fold. Swanson *et al.* (2002) defined by  $\gamma$  the ratio of the diffusion coefficient in white to that in grey matter, that is  $\gamma = D_w/D_g$ . An average diffusion coefficient for the entire brain can be defined as the diffusion coefficient in white matter times the volume fraction of brain that is white matter plus the diffusion coefficient in grey matter times the volume fraction of brain that is grey matter. The figures from the brain web database give the fraction of grey matter as 0.5723 and of white matter

as 0.4277 (Collins *et al.* 1998). So an average diffusion is given by

$$D_{average} = 0.5723D_g + 0.4277D_w = D_g(0.5723 + 0.4277\gamma) \\ \Rightarrow D_g = D_{average} / (0.5723 + 0.4277\gamma)$$

Swanson *et al.* (2002a) took as a typical average diffusion,  $D_{average} = 3.9 \text{ mm}^2/\text{month}$  so the diffusion in grey matter from the last equation gives  $D_g = 3.9 / (0.5723 + 0.4277\gamma) \text{ mm}^2/\text{month}$ . They evaluated survival time as a function of  $\gamma$  for a frontal tumour where it is mainly white matter and in the thalamic region where it is mainly grey matter.

Simulations of an anatomically correct brain highlights the problems with current imaging techniques. Figure 1 is a computed solution of equation (2) which shows the detectable tumour at death and the spread of the tumour cells beyond what can be detected by the most accurate current CT or MRI imaging techniques. Simulations of the model thus greatly enhance current imaging techniques to whatever level of cancer cell density is required.



**Figure 1** Computed solutions of equation (1) in a three dimensional anatomically accurate brain. These show the horizontal section of the virtual human brain through the site of the original tumour (+ in (a), \* in (b)). The left image in each is the tumour at diagnosis while the right image is the same tumour at time of death. The thick black contour defines the edge of the tumour that can be detected by enhanced CT. The blue contours outside this black line represent cancer cell densities peripheral to the imaging limits. (a) Tumour in grey matter: the time from diagnosis to death is approximately 256 days. (b) Tumour in white matter: the time from diagnosis to death is approximately 158 days. (Figures extracted from Swanson *et al.* 2002a).

The model described here has been used to quantify the effect of different treatment efficacies prior to their use. Incorporating periodic chemotherapy was studied by Tracqui *et al.* (1995) and Swanson *et al.* (2002b). The model used was equation (1) with a further (negative) term on the right hand side which quantifies the periodic reduction in growth as a consequence of the chemotherapy. Incorporating subtotal and total tumour resection in patient survival was considered by Woodward *et al.* (1996). This involves visually excising a given volume of the tumour in the model simulations. The predictions compared well with the data of Kreth *et al.* (1993). The modeling study (Woodward *et al.* 1996) predicted patient survival rates which, considering the basic aspect of their model, compared surprisingly accurately with the extant data at the time and recently published by Ramakrishna *et al.* (2010). Incorporating radiation treatment was also considered in the model and it has been used by Rockne *et al.* (2010) in the clinical study of 9 patients. A full review and how such treatments are incorporated are given by Murray (2003).

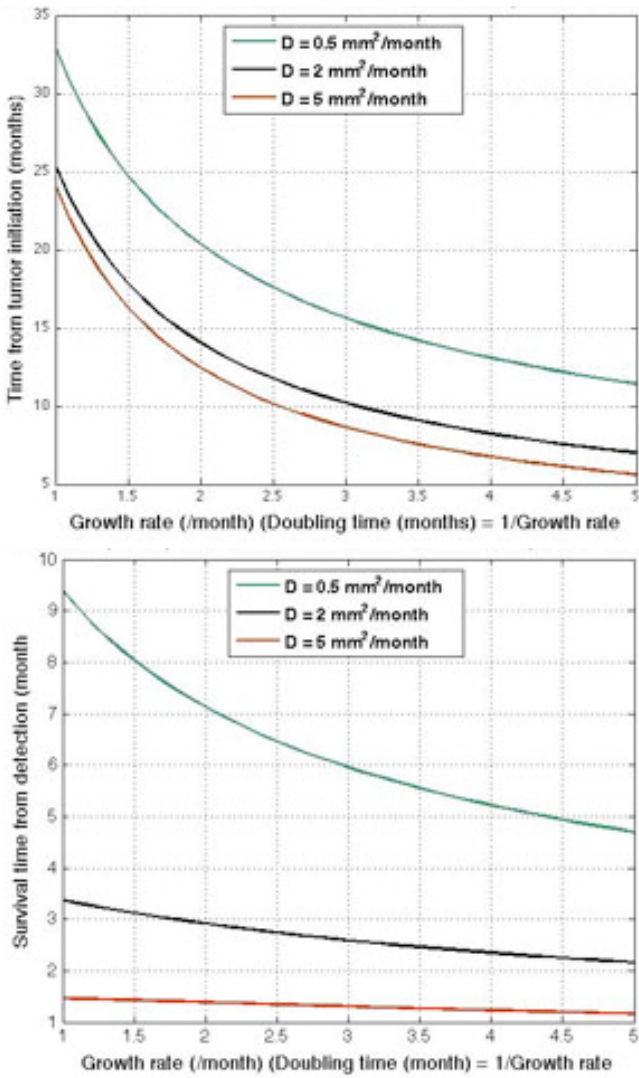
### Estimating the time from tumour initiation

An unsolved problem with all cancers is how to determine when a tumour started. In the case of glioma brain tumours detection is when the tumour volume is approximately equal to an equivalent sphere of radius 3cm in diameter but this also depends on the imaging technique used and where the tumour is in the brain. With the increasing discussion and justifiable concern of the possible increase in brain tumours as a consequence of the ever expanding use of cell phones it is inevitable that serious clinical studies will be carried out in the relatively near future. The paper by Tafforeau *et al.* (2004) clearly demonstrates the serious effect cell phone radiation has on plant growth. They showed that a single 2 hour exposure to radiation emitted at 105 GHz from a (GSM) cell phone resulted in considerable growth deformity. The *Journal of the American Medical Association* article by Volkow *et al.* (2011) reports on an increase in brain glucose in the region closest to the antenna. To date no study has definitively stated that brain tumours can arise from prolonged use of cell phones (I personally believe that there will be an increase in tumour incidence.)

Irrespective of the possible cell phone use connection, knowing when a tumour actually started is useful information which could possibly provide clues and pose relevant questions in any major clinical study.

With the increasing use and the quantitative clinical confirmation of many of the predictions of the model discussed here and in numerous publications since it was first introduced, it is reasonable to use it to obtain estimates of brain tumour initiation times. As a first approximation expression (6) gives the radius of the tumour and its velocity as a function of the diffusion coefficient and growth rate but for large times, mainly from when the tumour is first detectable, that is when it has an equivalent spherical volume of at least diameter 3cm. This however is only valid for sufficiently large times and although useful for calculating approximate life expectancy it is insufficiently accurate to back extrapolate to when the tumour started: it significantly underestimates the time.





**Figure 2** (top) This shows the survival time from initiation for a typical range of diffusion coefficients while (bottom) shows the survival time from detection at an equivalent tumour diameter of 3cm to death at a tumour diameter of 6cm.

We can obtain a considerably more accurate estimate using the exact solution (4) for the cell concentration as a function of time. If  $c_1(r, t)$  is the outermost cancer cell density level of detection when the tumour has an equivalent sphere radius of  $r$  then the time it takes for a density of  $N$  cells to grow and diffuse is given by the solution of (5) for given  $r$ ,  $c_1$ ,  $N$ ,  $D$  and  $\rho$ , namely the value of  $t$  such that

$$2t\sqrt{D\rho}\sqrt{1 - \frac{1}{\rho t} \log\left(\frac{c_1}{N}(4\pi Dt)^{3/2}\right)} - r = 0 \quad (9)$$

There is no analytical solution of this equation but it is possible to use MATLAB to obtain the value for  $t$  for given  $r$ ,  $c_1$ ,  $N$ ,  $D$  and  $\rho$ . This gives the time to initiation for all radii  $r$ , not only the radius at the smallest detection but whenever the tumour is first observed. It also gives a more accurate

estimate for the survival time by assigning the radius to be 3cm. We do not know how many cancer cells are required before they start to diffuse nor an accurate value for the detection level  $c_1$ . By way of example we chose  $c_1/N$  in (9) to be 80,000. Figure 2 illustrates the times from initiation for a typical range of growth rates and diffusion coefficients.

From Figure 2 the effect of higher growth rates play a smaller role than diffusion variability while at low growth rates the interplay between growth and diffusion is more complex.

## References

- Burgess PK, Kulesa PM, Murray JD, Alvord Jr EC (1997) The interaction of growth rates and diffusion coefficients in a three-dimensional mathematical model of gliomas. *J Neuropathol Exp Neurol* 56: 704–713.
- Collins DL, Zijdenbos AP, Kollokian V, Sled JG, Kabani NJ, Holmes CJ, Evans AC (1998) Design and construction of a realistic digital brain phantom. *IEEE Trans Med Imag* 17: 463–468.
- Cruywagen GC, Woodward DE, Tracqui P, Bartoo GT, Murray JD, Alvord Jr EC (1995) The modeling of diffusive tumors. *J Biol Systems* 3: 937–945.
- Kreth FW, Warnke PC, Scheremet R, Ostertag CB (1993) Surgical resection and radiation therapy versus biopsy and radiation therapy in the treatment of glioblastoma multiforme. *J Neurosurg* 78: 762–766.
- Murray JD (2002). *Mathematical Biology: I. Introduction*. (3rd edition, 3rd printing 2008) Springer, New York.
- Murray JD (2003) *Mathematical Biology II Spatial Models and Biomedical Applications* Springer, New York
- Ramakrishna R, Barber J, Kennedy G, Rizvi Win RH, Ojemann GA, Berger MS, Spence AM, Rostomily RC (2010) Imaging features of invasion and preoperative and postoperative tumor burden in previously untreated glioblastomas: Correlation with survival. *Surg. Neurol. Int.* 1:40.
- Rockne, R, Rockhill JK, Mrugala M, Spence AM, Kalet I, Hendrickson K, Lai A, Cloughesy T, Alvord Jr EC, Swanson KR (2010). Predicting the efficacy of radiotherapy in individual glioblastoma patients *in vivo*: a mathematical modeling approach. *Phys. Med. Biol.* 55 :3271-3285.
- Silbergeld DL, Rostomily RC, Alvord Jr EC (1991) The cause of death in patients with glioblastoma is multifactorial: Clinical factors and autopsy findings in 117 cases of supratentorial glioblastoma in adults. *J Neuro-Oncol* 10: 179–185.
- Swanson KR, Alvord Jr EC, Murray JD (2000) A quantitative model for differential motility of gliomas in grey and white matter. *Cell Prolif* 33: 317-329.
- Swanson KR, Alvord Jr EC, Murray JD (2002a) Virtual brain tumours (gliomas) enhance the reality of medical imaging and highlight inadequacies of current therapy. *Br. J. Cancer* 86:14-18.
- Swanson KR, Alvord Jr EC, Murray JD (2002b) Quantifying the efficacy of chemotherapy of brain tumours with homogeneous and heterogeneous drug delivery. *Acta Biotheoretica* 50:223-237.
- Tafforeau M, Verdus M-C, Norris V, White GJ, Cole M, Demarty M, Thellier M, 1, Ripoll, C. (2004) Plant sensitivity to low intensity 105 GHz electromagnetic radiation. *Bioelectromagnetics* 25:403-407
- Tracqui P, Cruywagen GC, Woodard DE, Bartoo GT, Murray JD, Alvord Jr EC (1995) A mathematical model of glioma growth: the effect of chemotherapy on spatio-temporal growth. *Cell Prolif* 28: 17–31.
- Woodward DE, Cook J, Tracqui P, Cruywagen GC, Murray JD, Alvord Jr EC (1996) A mathematical model of glioma growth: the effect of extent of surgical resection. *Cell Prolif* 29: 269–288.
- Volkow ND, Tomasi D, Wang G-J, Vaska P, Fowler JS, Telang F, Alexoff, D, Logan J, Wong C (2011) Effects of Cell Phone Radiofrequency Signal Exposure on Brain Glucose Metabolism. *J. Amer. Med. Assoc.* 2011;305(8):808-813.

# Prospects for Machine Embryogenesis

Jordan Pollack<sup>1</sup>

<sup>1</sup> DEMO Laboratory  
Brandeis University  
Waltham, MA, USA  
pollack@brandeis.edu

## Abstract

In Nature, the embryogenesis process proceeds from a single fertilized cell through division, migration, specialization and apoptosis. Although a lot is known about development, we still have a long way to go from theories of pattern formation towards understanding the intelligence within an unsupervised manufacturing process which robustly assembles complex biological forms.

Our approach has been to co-evolve bodies and brains in simulation and then convert them into reality using commercial manufacturing technology. I will review several generations of robots which were automatically designed using co-evolutionary techniques. The goal has been the fully automated design and construction of artificial lifeforms.

The first generation was based on genetic programming and a simulation of LEGO rod adhesion. The second generation used direct evolution on a iterative simulation of truss structures and used 3D printing for the output. A third generation was based on generative representations using L-systems.

In each of these cases, we assumed a perfect factory which could accept an evolved specification and then manufacture the desired result. In reality, there is no perfect factory, except for the science fiction Star Trek replicator. All manufacturing and assembly systems are subject to error. Each primitive manufacturing action results not in a deterministic new state, but a probability distribution of outcomes.

In later work, we replaced the idea of a perfect factory with one subject to noise and error. Even the smallest bit of error ruins the outcome of deterministic construction plans. We first evolved construction plans which could overcome errors through redundancy, and then this led to a new model for machine embryogenesis as a process which continuously optimizes assembly processes in a game against Nature.



## A review of one approach to bottom up assembly of minimal life

Steen Rasmussen<sup>1,2</sup>, Pierre-Alain Monnard<sup>1</sup>, Martin Hanczyc<sup>1</sup>, Anders Albertsen<sup>1</sup>, James Boncella<sup>3</sup>, Eva Bonzli<sup>1</sup>, Filippo Caschera<sup>1</sup>, Mark Dorr<sup>1</sup>, Harold Fellermann<sup>1</sup>, Maik Hadorn<sup>1</sup>, Wendie Jørgensen<sup>1</sup>, Philipp Löffler<sup>1</sup>, Sarah Maurer<sup>1</sup>, Kent Nielsen<sup>1</sup>, Pernille Pedersen<sup>1</sup>, Carsten Svaneborg<sup>1</sup>, Michael Wamberg<sup>1</sup>, Rafal Wieczorek<sup>1</sup>, Hans Ziock<sup>3</sup>

<sup>1</sup>Center for Fundamental Living Technology, University of Southern Denmark

<sup>2</sup>Santa Fe Institute, New Mexico, USA

<sup>3</sup>Los Alamos National Laboratory, New Mexico, USA

steen@lanl.gov

### Abstract

When seeking to assemble minimal life from the bottom up in wet carbon chemistry, the critical properties of life apparently emerge from the interconnected functions of three subsystems: information, metabolism and container. Such interconnected supramolecular systems, so-called protocells, are under the right circumstances able to mimic the main functions of a living cell although in a very simplified manner<sup>1</sup>.

Seeking to create minimal life from the top down leads us to a somewhat different picture, where construction of synthetic / streamlined genomes become the critical scientific issue<sup>2,3</sup>. How to integrate the knowledge we obtain from the top down- and the bottom up approaches is a great challenge for our and related communities<sup>4,5</sup> and a good problem to discuss at this meeting.

In technical terms, our bottom up team explores ruthenium-based photocatalysis as metabolism, fatty acids vesicles, oil droplets and reverse micelles as containers and lipophilic XNA as minimal informational systems<sup>6,7</sup>. Based on our experimental, computational and theoretical work we review protocell feeding, growth, division, motility, and information controlled metabolic production of containers<sup>8,9,10,11</sup>.

Finally, we demonstrate preliminary integration of biochemical- and microelectromechanical (MEMS) systems where life-like information processing and material production occur and interact in different media<sup>12,13</sup> and as such form an exciting frontier for the study of artificial life.

<sup>1</sup> Rasmussen S, et al., *Protocells: Bridging nonliving & living matter*, MIT Press, 2009

<sup>2</sup> Gibson DG, et al (2010) Creation of a bacterial cell controlled by a chemically synthesized genome. *Science* doi: 10.1126/science.1190719

<sup>3</sup> Rasmussen S, Life after the synthetic cell – Bottom up will be telling more (2010), *Nature*, 465422a, May 20

<sup>4</sup> Sunami T, et al., (2010) Detection of association and fusion of giant vesicles using a fluorescence-activated cell sorter, *Langmuir* 26: 15098

<sup>5</sup> Porcar M, et al., (2011), Ten grand challenges for synthetic life, to appear in *Synthetic Biology*.

<sup>6</sup> Rasmussen S, et al., (2003) Bridging nonliving and living matter, *Artificial Life* 9; 269

<sup>7</sup> Rasmussen S, et al., (2004) Transitions from nonliving to living matter, *Science* 303: 963

<sup>8</sup> Fellermann H, et al., (2007) Life-cycle of a minimal protocell – A dissipative particle dynamics study, *Artificial Life* 13; 319

<sup>9</sup> DeClue M, et al., (2009) Nucleobase mediated, photocatalytic vesicle formation from ester precursor molecules, *JACS* 131 931

<sup>10</sup> Toyota T, et al., (2009) Self-propelled oil droplets consuming “fuel” surfactant. *JACS*

<sup>11</sup> Maurer S, et al., (2011) Interactions between catalysts and amphiphilic structures and the implications for a protocell model. *Chem Phys Chem* 12; 828

<sup>12</sup> <http://www.fp7-matchit.eu>

<sup>13</sup> McCaskill, p. 253, in *Protocells: Bridging nonliving & living matter*, eds Rasmussen S, et al., MIT Press, 2009

# Synthetic Biocomputation: the possible and the actual

Ricard V. Solé<sup>1,2,3</sup> and Javier Macia<sup>1,2</sup>

<sup>1</sup> ICREA-Complex Systems Lab, Universitat Pompeu Fabra, PRBB-CEXS. Dr Aiguader 80, 08003 Barcelona, Spain

<sup>3</sup> Santa Fe Institute, 1399 Hyde Park Road, 87501 Santa Fe, New Mexico, USA

<sup>2</sup> Institut de Biologia Evolutiva, UPF-CSIC, Barcelona  
ricard.sole@upf.edu, javier.macia@upf.edu

## Abstract

Computation is defining trait of biological systems and a broad framework that captures the complex adaptive nature of molecules, cells and organisms. Computation is also at the core of the genotype-phenotype mapping, since it provides a natural framework to define function in a self-consistent way. The study of existing biological systems (from signalling cascades to ant colonies or brains) as well as the evolution of synthetic *in silico* networks performing computations reveals a number of nontrivial patterns of organization, sometimes in clear conflict with standard view of engineering or optimization. In spite of our increasing knowledge, there is a lack of a theoretical framework where computation and its possible forms is integrated within a general picture. Synthetic biology provides a new avenue where engineered molecular circuits can be implemented to perform non-standard computations. Here we review recent advances in the domain of multicellular synthetic computing and suggest a potential morphospace of computational systems including both standard and non-standard approximations.

## Introduction

Computation in nature is a fascinating and yet difficult topic. Biological systems perform computations as they gather information and process it in order to respond to environmental cues. Computation is in fact one formal way of capturing functionality in a well defined fashion (1), (2). Computation has also become a key aspect within the emergent field of synthetic biology (for a recent review, see (19)). This field allows to construct completely new molecular and cellular structures able to perform artificial computations (3).

Cells can be engineered in order to behave as autonomous, potentially programmable computing devices. These biocomputing devices would be able to perform complex tasks and designed for a wide range of applications, including bioremediation, food production or biomedicine (4). How to make these systems reusable and scalable is a major problem, but new approaches involving non-standard forms of computing have been able to overcome some key difficulties (5). They define novel ways of computing using living matter and suggest potential scenarios to outline a general framework to unify the landscape of computational structures, both in the natural and artificial realms.

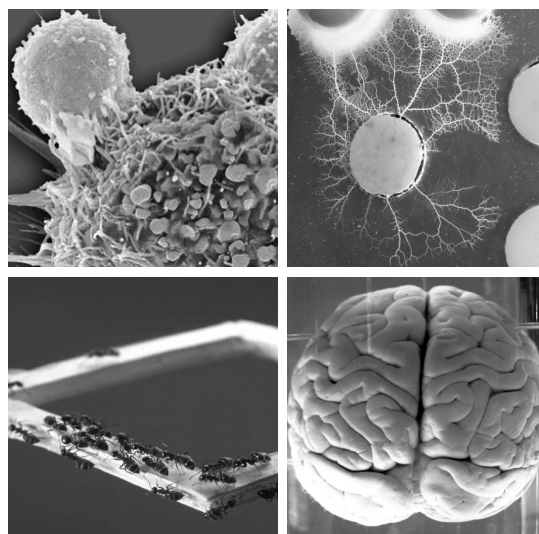


Figure 1: Computation occurs in natural systems in many different systems and spanning multiple scales. This include immune networks, social insect colonies, brains or some social amoebae.

In order to use computation as a unifying framework where biological complexity and its evolutionary dynamics can be suitably integrated, some formalism is needed. One possibility is to consider classical models of computation. Turing's formalization of computations in terms of machines with a number of internal states provides a powerful framework where -in principle- any potential form of computation could be described (6). The fact that some particular macromolecular systems, such as ribosomes act pretty much as Turing-like nanomachines (reading a "tape" defined by the messenger RNA, creating an output chain of aminoacids and starting and ending the process by means of detecting given sequences) seems to support this picture. Such avenue has been successfully taken by some researchers (7) proving the viability of making molecular computations close to finite automata. However, as pointed out by Melanie Mitchell (8) there is a range of biological systems, from immune net-

works to ant colonies or even plants, where computations occur and yet seem to escape from being fully captured by classical, Turing-like formal approaches to computation.

The special features shown by information-processing systems in biology have been recognized for decades. Many of them have to do with special ways of treating given computational tasks in a parallel way and using the internal dynamical features characteristic of each system. Task allocation in ants, for example, can be favoured in some cases by means of colony-level oscillations which seem in principle inappropriate for dealing with colony needs. Simple models of ant dynamics based on a neuron-like mapping between ant states and formal neurons have been very useful in this context. In particular, it has been shown that oscillations actually favour an optimal task fulfilment that is not possible if a constant, average activity level were at work (9), see also (10).

Similarly, other properties exhibited by complex biological machines strongly depart from standard engineering-based principles. One such principle is the robust behavior based on redundancy. Here two identical components of the system making the same function can replace each other in case of failure. Redundancy is thus the intuitive (although sometimes expensive) solution to the problem of failure. However, it has been shown that in many cases (may be in most cases) robust behavior is not obtained from redundant structures. Instead, it seems to be a consequence of so called degeneracy (11), (12), (13). It can be defined as the capacity of elements of a given system that are structurally different to perform the same function or yield the same output. This ubiquitous feature appears to be present in many different systems and scales. Modeling *in silico* evolved circuits performing computations under selection for robust behavior (14) reveal that robustness is achieved through degeneracy, but the underlying mechanistic explanation escapes from our intuition. Degeneracy implies a novel concept beyond standard engineering, suggesting that new forms of thinking might be required.

How can we go beyond the limits imposed by real systems, which are the result of evolution and might be difficult to fully characterize? Similarly, how can we test existing theories and try novel ones if they are sometimes difficult to compare with their real counterparts? The field of synthetic biology seems to provide the best scenario for designing novel computational systems *in vivo* whereas non-standard forms of computation are used as alternatives to engineering-inspired metaphors. Here we present some of these results and suggest a potential framework to define a space of computational designs that includes existing natural and artificial systems as well as engineered, artificial ones.

### Logic gates from gene circuits

One way of creating synthetic biological circuits performing predefined logic operations is based on engineering genetic

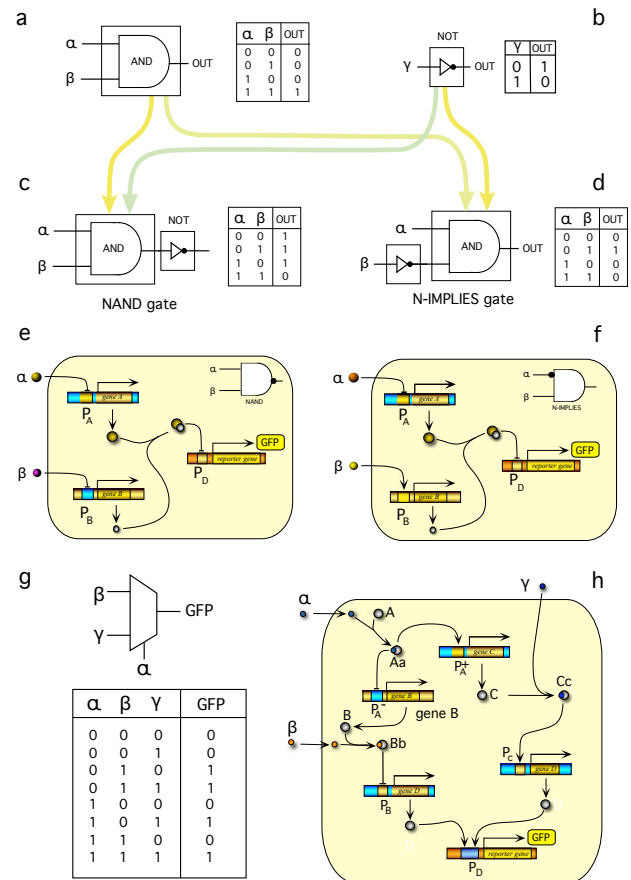


Figure 2: Logic gates and switches of different types can be obtained by engineering cellular and/or molecular systems. Examples would include (a) the AND and (b) the NOT gates, from which a NAND gate (c) or a N-IMPLIES gate (d) can be obtained through combination. In (e) and (f) we illustrate these two examples through a hypothetical gene regulatory system (the inset pictures are the compact representation of the gates).

regulatory systems. In figure (2) we show some examples of logic gates that can be implemented by using available genetic components and their interactions. Such circuits are obtained by means of standard genetic engineering techniques and the components can actually come from different, completely unrelated species, which can mix together genes from viruses, bacteria or mammals. Typically these engineered circuits are built within plasmids, i. e. closed chains of DNA defining genetic information physically separated from the chromosomal DNA. A different strategy involves using appropriate gene. In figure 2a we illustrate this by means of two basic examples and their genetic counterparts (other implementations are also possible). In our example (e) the NAND gate is obtained by using a molecular complex formed by two different proteins which repress the

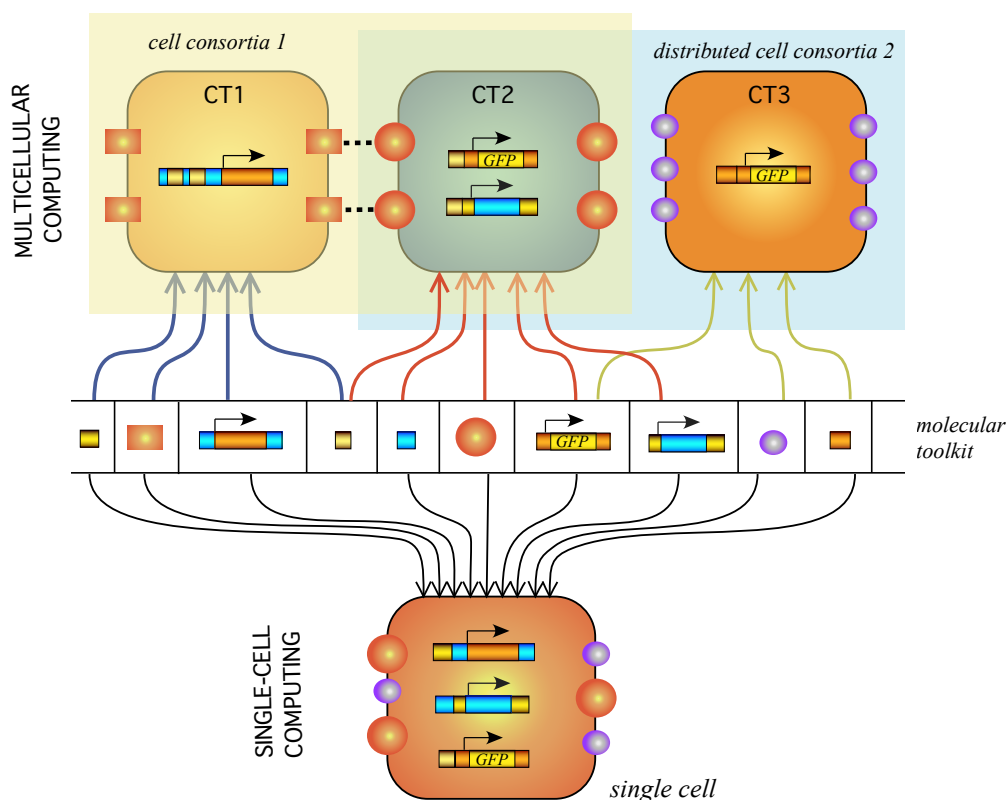


Figure 3: A genetic toolkit (central line) can be used to engineer cellular computations. The toolkit might include all sorts of regulatory elements and reporters. The standard approach is putting together within a single cell, where all regulations take place. Alternatively, a library of different engineered cells can be created, thus defining a cellular consortia (top diagrams).

expression of a so called reporter gene (here GFP=green fluorescent protein) which generates, when activated, a fluorescent signal.

These examples illustrate the standard approach of electronic design based on combinatorial logic. In principle, every circuits could be designed in this way. However, a major difficulty emerges here: in electronics, every wire is defined in terms of a conducting piece of material, which is always the same. When dealing with cellular engineered systems, where molecules share the same medium where they are mixed, identity becomes a problem. In a cell, every wire needs to be a different molecule to properly connect different elements or cells. Because the liquid nature of the medium where computations need to occur, the spatial insulation of wires that is assumed in electronics is no longer satisfied. As a consequence, each wire needs to be implemented by using a different molecular carrier and the chemical diversity of constructs rapidly grows. This is illustrated in figure 1g-h by the so called multiplexor, widely used in electronic designs. This is a 3-input, one-output system where a given signal "selects" one of the two inputs. In principle a synthetic genetic network implementing a MUX circuit can be designed (an example shown in figure 1h) using a single cell

implementation. Although such circuit can be constructed (15) it is a hard task, with no hope of being re-used as part of a larger system (as it occurs in electronics).

To sum up, the combinatorial approach can lead to a nightmare when dealing with an experimental design, since the properties of each carrier and how it interacts with other can be very different and difficult to predict. Additionally, one goal of the field is to have engineered systems capable of extensive reuse of available parts in such a way that a LEGO-like system is at work. Both premises are basic requirements for reaching the computational complexity for achieving autonomous machines able to make decisions in a biological context. Only recently a general approach, based on engineering several cell types, has been successfully obtained.

### Cellular consortia: division of labor

One way of dealing with the wiring problem is considering alternative ways of avoiding the mixing of molecular carriers that seems inevitable within the cell cytoplasm. Spatial segregation of the basic components provides one easy way of dealing with computation avoiding molecular mixing. Although the explicit use of spatial locations is one possibility

(see for example (16) a simpler scenario involves using cellular consortia, namely a population of cells having different types of engineering designs. In such scenario, a library of different cell types, each one having a different subset of genetic components, is build out of a collection of molecular components. This is schematically displayed in figure 3, where we show three different ways of combining them within single or multicellular constructs.

Here different cell "types" are indicated as CT1, CT2, etc. A standard consortium (top left) is obtained by splitting some of the elements from the toolkit between cells. Communication is then also introduced, so that a sender and a receiver cell are usually designed, although feedbacks are also introduced in most designs. A reporter cell is present (here CT2) which will (1) or will not (0) express a target molecule. This type of consortium has been used in many different contexts. In particular, using two cell types it was possible to artificially recreate predator-prey systems (Lotka-Volterra dynamics), mutualistic ensembles (hypercycle-like systems) or parasitic organizations. Extensions of these include multispecies ecosystems where different groups of cells belonging to different kingdoms are involved (17). Once again, however, the resulting synthetic cells are hard to reuse to obtain other types of computations. An alternative approach requires breaking some predefined rules.

In any standard circuit design, the truth table defines the input-output relation between incoming sets of signals and the resulting outputs. The outputs are placed in given locations of the circuit and it makes sense that this is the case. Let us limit ourselves here to a single-output system. That means that there is an output unit where the final result of the information processing is released. What happens if we free ourselves from such (rather reasonable) assumption? The view of a computational device as being implemented by a circuit that clearly differentiates between input, processing and output units seems too obvious to replace it by some other paradigm. But there is actually one solution that emerges from not forcing that assumption to be true. Instead, more than a cell type is able to respond as output element.

## Distributed computation

Here we introduce our basic model approach to synthetic computation. We will use a Boolean approximation, thus confining our approximation to the digital domain. Our state space will be described by a set  $\Sigma = \{0, 1\}$ . Although this is in principle a limitation, many relevant cellular computations seem to take place by means of genetic switches. Such switches effectively define binary states with low and high levels of gene expression. A given functionality will be described as an input string  $\mathbf{I}$ , namely an element of

$$\Sigma^N = \underbrace{\{0, 1\} \times \dots \times \{0, 1\}}_n \quad (1)$$

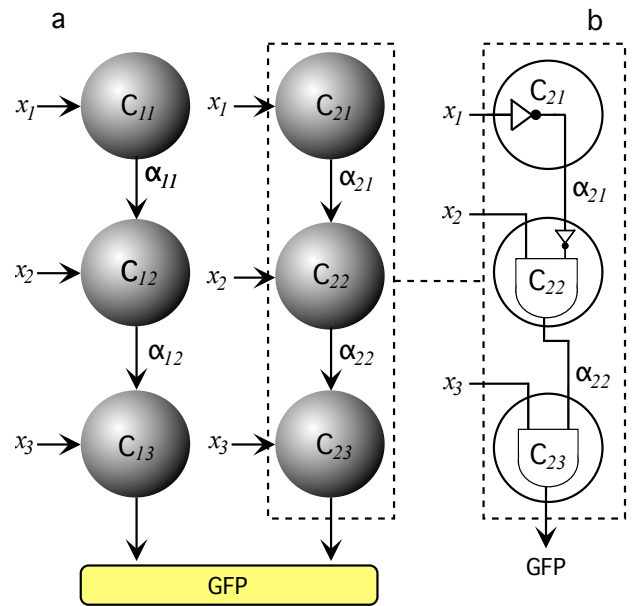


Figure 4: The general architecture used here to generate our circuits using distributed computation, as defined in the text. A feed-forward structure (a) is assumed as the basic scaffold, with a number of input signals that affect separated arrays of communicating cells (gray spheres). Each cell implements a given logic function (b). All columns end up in a reporter cell, but several reporter cells can be present, since each column is segregated from others, thus removing potential cross-talks.

It will indicate, in our framework, a string of absent (0) or present (1) chemical signals.

The functional trait to be implemented is formally defined as a Boolean function  $\phi_i$  with  $N$  input signals and a single output. Formally, this reads:

$$\phi_i : \Sigma^N \longrightarrow \Sigma \quad (2)$$

Two particularly relevant subsets of Boolean functions are the one input-one output gates, i. e. the set  $\mathcal{G}^{(1,1)} = \{NOT, Id\}$  (the negation and identity functions, respectively) and the  $2^4$  two-input logic gates defining the set  $\mathcal{G}^{(2,1)} = \{g_j\}$  where  $g_j$  is a mapping

$$g_j : \Sigma^2 \longrightarrow \Sigma \quad (3)$$

(represented by a simple table). Standard functions include OR, AND and their "inverse", i. e. NOR and NAND.

Our approach to a general design of complex computational circuits (5) is based on two general assumptions, to be translated into a basic circuit design (fig 4). First, we limit ourselves to a feed-forward network where each node is one type of engineered cell from the library. Each link means the existence of a molecular connection, i. e. a diffusible

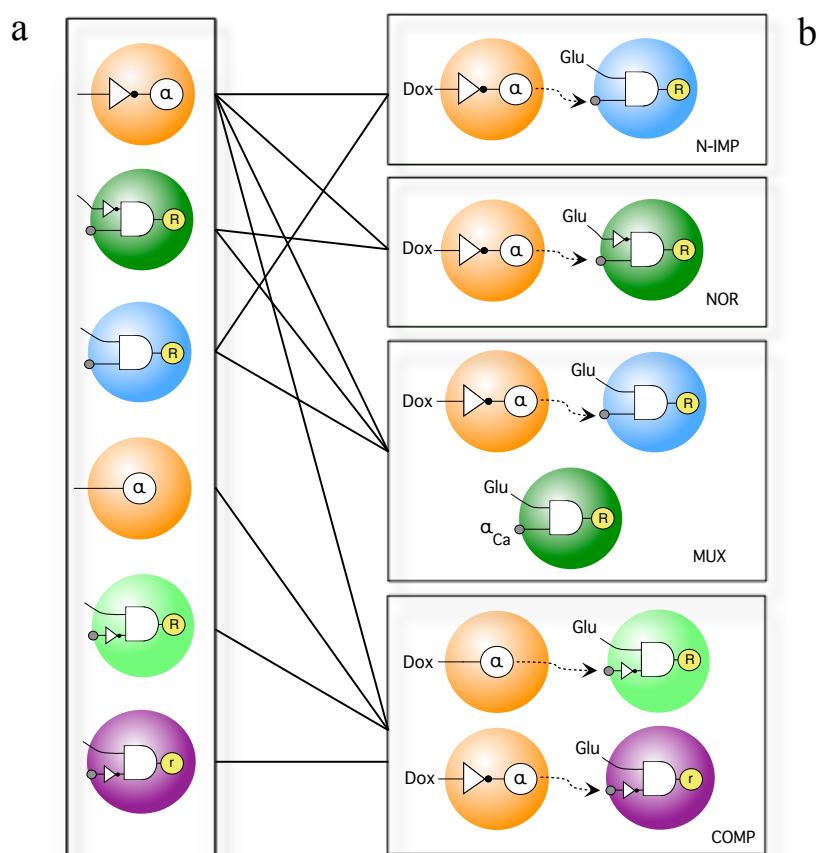


Figure 5: Combinatorial design of multicellular consortia using our distributed computation method. Here in (a) we sketch some basic engineered cell types (implemented on yeast cells) and several examples (b) of combinations performing given functions. Many other circuits can also be constructed using this library or small extensions of it.

wire molecule. Secondly, several cell types can incorporate the gene responsible for the output molecule (GFP). Once a given Boolean table is chosen, an evolutionary algorithm is applied to the basic wiring structure, which explores the landscape of potential networks implementing the desired function (3). The algorithm searches over the space of basic functions, wiring configurations and other constraints. Once a given network is found, standard rules of circuit minimization are applied in order to obtain the minimal circuit solving the problem by means of distributed computation.

What are the results of this method? Along with this evolutionary algorithm, the theoretical analysis demonstrates that it is possible to minimize the number of required cells and wires using the distributed output assumption combined with a small library of cells implementing only the AND and the inverted Implies gates (N-IMPLIES). Despite this combination of gates are not usually used in circuit designs they

define a functional complete set, i.e. any arbitrary Boolean function can be implemented only combining this two gates. In some embodiments, these gates can be simplified and replaced by the IDENTITY and the NOT gates respectively allowing for a circuit simplification. Furthermore, the wiring pattern of connections is restricted, i.e. different circuits can involve different number of cells and wires but all cells only respond to an external input and to single diffusible molecule acting as a wire according with the specific logic function implemented, i.e. AND or N-IMPLIES, independently on the circuit complexity.

Using yeast cells as the model organism to implement our cell library (following the theoretical predictions) it was possible to construct, by combining different cell types, all kinds of simple gates (figure 5) but also complex circuits. As an illustration of the enormous simplification of circuit complexity that is derived from our approach, in figure 5 we



can see that the MUX circuit can be obtained by combining three cells from the library (it can also be done with only two). Similarly, much more complex circuits, such as a binary adder (figure 5) was also obtained. As we can see from the two complex circuits, the assumption of distributed computation makes possible to actually split the circuit in different segregated (and thus disconnected) parts. Once again, this is in deep contrast with the standard view of electronics.

A final result concerns the predicted types of cell-cell interactions that is predicted by the evolutionary algorithm. Using the MUX circuit as the basic reference, we run the algorithm in such a way that many different circuits were obtained, all consistently implementing the multiplexer. In figure 6 we display a graph summarizing two relevant pieces of information. The nodes are the basic gates implemented by individual cells. Their size here is proportional to their frequency in the evolved circuits. We can see that there are wide differences between different logic components. Secondly, the weighted links between different gates indicate how frequently two given gates appeared connected within a given solution. The resulting network illustrates once again the nonstandard character of our solutions. The first lesson is that, although it is known that NOR and NAND gates could be in principle used as the single logic elements to implement any logic circuit (18) this solution is largely ignored by the algorithm. Secondly, the N-IMPLIES function, which was successfully used in (5) seems to be a key component in most solutions. Since the N-IMPLIES gate is not a standard component in electronics but seems to be very important here, this suggests that some design principles used in synthetic biology might need to be revisited.

The potential power of distributed computation as described above is illustrated by noticing that even a small number of engineered cell types makes possible to create hundreds of synthetic circuits (5) and thus a huge potential array of functions. Adding wires makes the combinatorial power of the system to rapidly increase in orders of magnitude the number of potential circuits, which are easily achieved thanks to the enormous capacity for tinkering and combination.

## Discussion

Synthetic biology has been rapidly gaining relevance and potential as novel techniques are getting incorporated to the field and new applications start to emerge (?), (20), (21). Our view of the area in terms of computation is simply a way of addressing the combinatorial potential of functional circuits in a very broad way. Such view allows to properly address some of the key problems in the field, namely wiring constraints and real combinatorial design. Our recent work indicates that by removing the assumption of specified output units, by allowing the output to be distributed over multiple cell types, low-wiring, combinatorial circuits can be obtained.

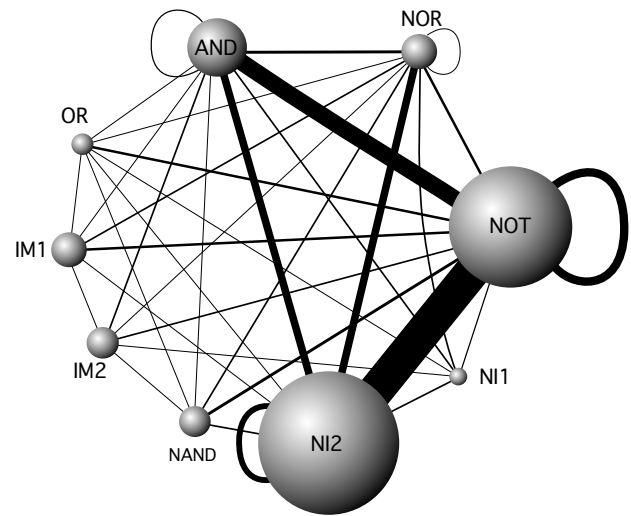


Figure 6: The weighted network obtained for many different evaluations of the evolutionary algorithm searching for multicellular MUX networks.

The previous results are encouraging in two different ways. On the one hand, given the truly combinatorial potential of the method, hundreds of possible synthetic designs can now be constructed. The method allows to predict possible ways of building minimal circuits and thus adapt the required result to experimental constraints. But it also opens an interesting framework to approach more general questions. Our method shows that an unexpected way of solving computational problems can be obtained.

The resulting solutions are counterintuitive and reveal an alternative form of actually achieving the right computation through cellular consortia that can be disconnected into several pieces. Moreover, the results might be more general. For convenience, we have presented our work in terms of cellular consortia, where the basic, spatially defined units are cells. But it might well be the case that other scenarios, such as sub-cellular structures, also fit within our framework. Different cellular compartments could in principle perform parts of the computational processing required to implement a given function in a distributed manner. Since biology tends to make possible everything that can be imagined under reasonable terms, we predict that the kind of computations presented here are likely to be found in living systems.

It is also interesting to notice that reliable computation at low wiring cost has been achieved through a method where autonomous parts emerge as part of the solution. Given a function to be implemented, the architecture of the resulting design involves different parts contributing to the overall computation but essentially independent. This result suggests that the evolved circuits might actually display a high degree of robustness, and preliminary results seem to con-

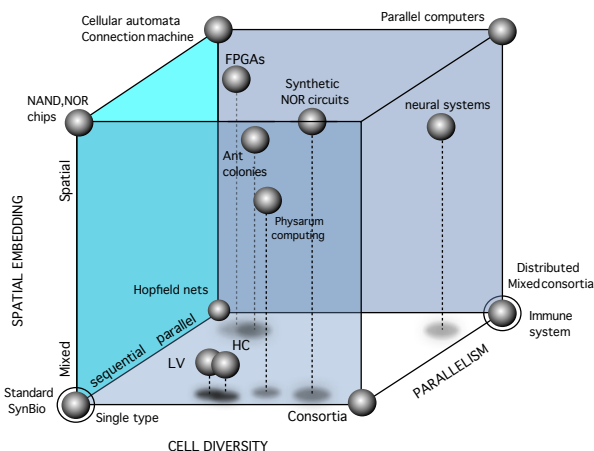


Figure 7: A toy space of computations where the three axes include the presence of spatial segregation, degree of multicellularity and to what extent the computation is sequential or parallel. Several well known examples are located at roughly representative locations. Here LV= Lotka-Volterra synthetic ecosystem, HC=mutualistic synthetic system, NAND, NOR chips: small chips constituted by only NAND and NOR gates, widely used as basic building blocks in many electronic designs; FPGAs: field programmable arrays.

firm this point.

Finally, our results also introduce an additional layer of complexity within biological computation. If we define an imaginary, qualitative space of computational structures where the number of different cells and the presence of space define two axes, a third one would be the relative importance of distributed computation as defined here (in terms of the output). A tentative (and by no means exhaustive) picture of this space is provided in figure 7. In this space, we have allocated different known systems that differ in their computational power and how it works. Our distributed computation approach defines, a corner of this diagram (encircled, right sphere). The three axes are intended to capture three relevant features of computational systems. These are: (a) degree of parallelism, (b) diversity of units involved (cells for our engineered systems but can be ant castes or electronic components in other contexts) and (c) spatial embedding, meaning how relevant is the spatial distribution of the agents while performing computations.

This is a largely unexplored space, where spatial degrees of freedom can help to further simplify our implementation and simultaneously increase our combinatorial power (unpublished results). Moreover, it is possible to show that a limit case of our implementation, where all cells in the consortia are actually disconnected among them, can successfully be implemented too. Many open questions emerge from this work, but it also provides an elegant and promising

scenario where many relevant questions will be testable, including potentially unexplored forms of computation, their robustness and evolution.

## Acknowledgements

We would like to thank the members of the Complex Systems Lab as well as to F. Posas, L. de Nadal, N. Conde and S. Regot for useful discussions. RVS also thanks his colleagues S. Kauffman, M. Mitchell, D. Krakauer, C. Moore and JF Sebastian for interesting comments. This work has been supported by grants from the Ministerio de Ciencia y Innovación (BIO2009-07762) and the James McDonnell Foundation; the CELLCOMPUT (FP6) project, The Santa Fe Institute. Special thanks are given to the Fundación Marcelino Botín (FMB).

## References

- Amos, M. (2004). Cellular Computing. *Oxford University Press*, New York.
- Regev, A. and Shapiro, E. (2002). Cellular abstractions: Cells as computation. *Nature*, 419:343.
- Macia, J., Posas, F. and Sole, R.V. (2011). Synthetic biological computers. *Trends in Biotechnology*, In press.
- Benenson, Y. (2009). Biocomputers: from test tube to live cells. *Mol. BioSyst*, 5:675-685.
- Regot, S., Macia, J., Conde, N., Furukawa, K., Kjellen, J., Peeters, T., Hohmann, S., de Nadal, E., Posas, F. and Sole R.V. (2011). Distributed Biological Computation with Multicellular Engineered Networks. *Nature*, 469:207-211
- Turing A.M. (1937). On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.*, 42:230-265
- Benenson, Y., Paz-Elizar, T., adar, R., Keinan, E., Livneh, Z. and Shapiro, E. (2011). Programmable and autonomous computing machine made of biomolecules. *Nature*, 414:430-434.
- Mitchel, M. (2011). Ubiquity symposium: Biological Computation. *Volume 2011 Issue February*, ACM New York, USA.
- Delgado, J., and Solé, R.V. (2000). Self-synchronization and taks fulfilment in ant colonies. *J. Theor. Biol.*, 205:433-411
- Solé, R.V. and Delgado, J. (1996). Universal Computation in Fluid Neural Networks. *Complexity*, 2:49-56.
- Tononi, G., Sporns, O. and Edelman, G.M. (1999). Measures of degeneracy and redundancy in biological networks. *Proc. Natl. Acad. Sci. USA*, 96:3257-3262.
- Edelman, G.M. and Gally, J.A. (2001). Degeneracy and complexity in biological systems. *Proc. Natl. Acad. Sci. USA*, 98:13763-13768.
- Wagner, A. (2005). Robustness and evolvability in living systems. *Princeton U. Press*. Princeton.
- Macia, J. and Sole, R.V. (2009). Distibuted robustness in cellular networks: insights from evolved diital circuits. *J. R. Soc. Interface*, 442:259-264.



- Moon, T.S., Clarke, E.J., Tamsir, A., Clark, R.M., Eanes, M., Korf, T. and Voigt C.A. (2011). Construction of a genetic multiplexer to toggle between chemosensory pathways in *Escherichia Coli*. *J Mol Biol*, 406:215-27.
- Tamsir, A., Tabor, J.J. and Voigt, C.A. (2011) Robust multicellular computing using genetically encoded nor gates and chemical 'wires'. *Nature*, 469: 212-5.
- Weber, W., Daoud-El Baba, M. and Fussenegger, M. (2007). Synthetic ecosystems based on airborne inter- and intrakingdom communication. *Proc. Natl. Acad. Sci. USA*, 104:10435-10440.
- Enderton, H. (2001). A mathematical introduction to logic. Harcourt Academic Press.
- Purnick, P.E. and Weiss, R. (2009). The second wave of synthetic biology: from modules to systems. *Nature Reviews Molecular Cell Biology*, 10:410-422.
- Abelson, H., Allen, D., Coore, D., Hanson, C., Homsy, G., Knight, T.F., Nagpal, R., Rauch, E., Sussman, G.J. and Weiss R. (2000). Amorphous computing. *Communications ACM*, Volume 43, Number 5.
- Weiss, R., Basu, S., Hooshangi, S., Kalmbach, A., Karig, D., Mehreja, R. and Netravali, I. (2003). Genetic circuit building blocks for Cellular Computation, Communications, and Signal Processing, *Natural Comput.*, 2:47-84.

# ALife: a Tension Between Biology and Software Engineering

Susan Stepney<sup>1</sup>

<sup>1</sup>Dept. Department of Computer Science  
University of York, UK  
susan.stepney@cs.york.ac.uk

## Abstract

Biological systems are insanely complicated. If we look at the details of plant growth, of the vertebrate adaptive immune system, of bacterial horizontal gene transfer (to pick three areas with which I have a soupçon of familiarity), it is *all* insanely complicated, on every level, from top to bottom. When a feature is this ubiquitous, it may just be necessary. It is at least telling us something important.

One of the guiding principles of ALife is studying, understanding, and creating life *from the bottom up*. Since our only current exemplar, biological life, is insanely complicated, even at the bottom, what does this tell us about *in silico* implementations?

Everything we are taught in Software Engineering is about reducing, constraining, containing, and managing complexity. Well-defined small stable interfaces. Formally specified requirements. Rigorous development of correct code. And all this known and documented before the code is deployed in the field.

Life however exhibits open-ended evolution, continual novelty: not only new organisms, but new species, new families, new phyla, new kinds of life. Evolution evolves. The code of life writes itself.

Object-oriented agent-based simulations running inside an evolutionary algorithm, no matter how bio-inspired the genetic operators, nor how bio-inspired the developmental stage, are *closed*. They cannot escape their small pre-specified box in possibility space. They cannot exhibit open-ended evolution. If we want life *in silico*, we have to allow the code to write itself.

I am not suggesting that we throw up our hands in despair, pour assembly language into a big bucket and just let it trample all over itself, in the hope that life will emerge after several billion CPU years. We can use bio-inspiration at the whole simulation level, to develop code that can self-adapt and self-modify in ways plausibly analogous to bio-evolutionary processes. Hand-in-hand with our sophisticated understanding of biology, we need to use more sophisticated computer science, including self-modification through computational reflection.

## The mechanics of shape change in the *Drosophila* embryo

Eric F. Wieschaus<sup>1,2</sup>, Adam C. Martin<sup>5</sup>, Bing He<sup>2</sup>, Matthias Kaschube<sup>3,4</sup>, Oleg Polyakov<sup>4</sup>

<sup>1</sup>Howard Hughes Medical Institute, USA

<sup>2</sup>Department of Molecular Biology, Princeton University, USA

<sup>3</sup>Lewis-Sigler Institute for Integrative Genomics, Princeton University, USA

<sup>4</sup>Joseph Henry Laboratories of Physics, Princeton University, USA

<sup>5</sup>Department of Biology, Massachusetts Institute of Technology, USA

efw@princeton.edu

### Abstract

With the first three hours of development, the *Drosophila* embryo establishes a precise pattern of transcription factors that divides the blastoderm into groups of cells destined to form different organs and tissues in the adult. Along the dorsal ventral axis, the first and perhaps most important of these cell fate decisions is the establishment of mesoderm controlled by expression of the Twist and Snail transcription factors. These cell fate decisions are immediately translated in changes in the shapes and physical properties of the 800 mesodermal cells and result in the formation of the furrow that translocates them to the interior. Although at the cellular level these changes involve a re-organization of the cytoskeleton, adhesion and motor activities to achieve distinct shape we are interested in the underlying physical parameters that govern behavior.

In my talk I will discuss the relationship between the initial transcription profiles and a novel pulsating reorganization of the Actin/Myosin cytoskeleton in the apical region of cells that will make the ventral furrow. We show that the resultant contractile pulses drive cell shape changes in the entire mesodermal primordium. The individual contractions appear to be unpolarized but they result in polarized wedge-like constrictions because global tension in the sheet is polarized along the AP axis. We analyze the force distributions in the mesodermal primordia using a combination of genetics and RNAi to lower adhesive strengths between cells, and laser dissections to locally disrupt the cytoskeleton.

We have developed analytical tools that allow tracking surface areas and volumes of all 800 mesodermal cells during the process of furrow formation. We find that cell volume is essentially constant during the process and that global cell shape changes are pulsed in synchrony with the Actin/Myosin contractions in the apical surface. We envision that force generated apically is transmitted over large distances by the non-compressible nature of the cytoplasm and suggest that similar mechanism that may underlie many morphogenetic movements.

# Effective Calibration of Artificial Gene Regulatory Networks

D. Agostini<sup>1</sup>, J. Costanza<sup>1</sup>, V. Cutello<sup>1</sup>, L. Zammataro<sup>2</sup>, N. Krasnogor<sup>3</sup>, M. Pavone<sup>1</sup>, and G. Nicosia<sup>1</sup>

<sup>1</sup>Dept. of Mathematics and Computer Science, University of Catania

{costanza, vctl, mpavone, nicosia}@dmi.unict.it

<sup>2</sup>Dept. of Translational Medicine, University of Milan, IRCCS Istituto Clinico Humanitas

luca.zammataro@humanitasresearch.it

<sup>3</sup>School of Computer Science, University of Nottingham

Natalio.Krasnogor@Nottingham.ac.uk

## Abstract

Knowing every single component of a given biological system is not enough to understand the complexity of the system but rather it becomes crucial to understand how these components interact with each others. It is not only important the knowledge of genes and proteins, but also knowing their structures and primarily the laws and parameters governing their dynamics, which is often unknown and impossible to measure directly. The *Gene Regulatory Networks* explain exactly how a genomic sequence encodes the regulation of expression of sets of genes, which progressively generate developmental patterns, and execute the construction of multiple states of differentiation. Their main aim is to represent the regulation rules underlying the gene expression. In this work we have designed the CMA-ES algorithm to infer the parameters in the S-system model of a gene regulatory network. This model is a well-known mathematical framework whose structure is rich enough to capture many relevant biological details, and it can model more complicated genetic network behaviour. CMA-ES has been compared against 7 state-of-the-art algorithms to evaluate its efficiency and its robustness. From a general point of view, it seems clear how CMA-ES is able to estimate in a better way the target parameters with respect to the state-of-the-art methods, either in terms of success rate or in terms of Euclidean distance. Finally, this research paper includes a study on the convergence of CMA-ES through Time-To-Target plots, which are a way to characterize the running time of stochastic algorithms; and a global sensitivity analysis method, the Morris algorithm.

## Introduction

In the past few years studying how a system interacts with the environment, or how simple components effect the global behaviour of a given system, or even how parts of a system interact with each others has been the main and most challenging issue in many research areas. Many problems in science and engineering are often hard to solve mainly because of the difficulty in understanding their indirect causes and effects, which are not related in an obvious way. Assessing all the single parts of a structure, or knowing all single components of a given system is not enough to determine and understand the complexity of system, although we need to know how these objects interact. It is also well known that

the information on the complex molecular features are contained in the genome of the organism, but is not clear what are the codes and mechanisms that translate the sequences into structures and functions. For example, from systems biology point of view, it is not only important a knowledge of genes and proteins, but it is of primary importance understanding their structures, dynamics, and how their parameters influence the global dynamics: such parameters are unknown and often impossible to measure directly. Moreover, studying dynamic properties of a biological system is not only very important to gain a deep understanding of biological processes, but also to develop efficient treatments against diseases. In systems biology *reverse engineering* the processes can be regarded as a central part of the discipline itself (Lee, 2005). Reverse engineering can be considered as a process from which is possible to infer structural and dynamics features of a given system from external observations and relevant knowledge. Thanks to that, today reverse engineering techniques play a central role in systems biology (Csete and Doyle, 2002; D’haeseleer et al., 2000). The main focus in reverse engineering field is the identification of genetic networks (Cho et al., 2007) in order to learn how transcription factors are connected to genes (the determination of the interactions between all genes and understanding of the regulatory networks are crucial to identify and develop novel drugs), and understand the gene expression profile that is a major issue in computational biology. In other words, reverse engineering can help us to answer questions as: (1) what are the functions of this gene? (2) which genes regulate this gene? (3) how several genes interact? (4) which genes are responsible for this disease? (5) which drugs will treat this disease? Of course, a method to interpret these answers is needed, in order to enhance our learning of living organisms. *Gene Regulatory Networks* (GRNs) explain exactly in which way genomic sequences encode the regulation of expression of sets of genes that progressively generate developmental patterns and execute the construction of multiple states of differentiation (Davidson and Levin, 2005). The main aim of GRN is to represent the regulation rules underlying the gene expression. Albeit the study of GRNs

nowadays is made easier thanks to the advances of new technologies however the solution to the problem is not trivial due to the enormously large space of the unknown parameters. In the last years several reverse engineering methodologies based on evolutionary algorithms have been presented (Kabir et al., 2010; Kikuchi et al., 2003; Noman and Iba, 2007), which are more suitable to effectively and efficiently reconstruct the networks in a given dynamic model. It is well known as evolutionary algorithms work better than standard methods when the problem to solve is nonlinear, and therefore or no solution is known a priori, or it is impossible to be analytically solved. The great advantage of the evolutionary approaches on these tasks is own their applicability to almost any models where mathematical analysis and reversing is unavailable or inefficient. A good comparative study among evolutionary algorithms in gene regulatory network can be found in (Schlitt and Brazma, 2007). In this research work, we present a new approach to infer parameters of a gene regulatory network from time-series gene expression data using *S-system* model (Irvine and Savageau, 1990). For this kind of task, one of the best population-based optimization algorithms has been used as learning paradigm: *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES) (Hansen and Ostermeier, 2001).

### The S-system model for gene expression

Developing accurate computational and mathematical models is needed to study the response of the gene regulation and the gene sets with respect to their specific dynamics (many important cell functions are largely determined by dynamic processes of biochemical networks). Therefore, using mathematical models for the analysis of metabolic and regulatory pathways may contribute to a better understanding of the behaviour of metabolic processes. These models, once built, can be used to predict the behaviour of the organism under certain conditions (Sirbu et al., 2010), it has been also postulated that, once inferred the basic mechanisms of life, it should be (theoretically) possible to create synthetic organisms (Barrett et al., 2006). Of course, the choice of the model to use depends by how much information we try to capture: more information a model trying to learn - more parameters need to be inferred - more complex becomes the model. Nowadays, there exist several types of models in literature that describe a gene regulatory network, as: *Boolean networks* (D’haeseleer et al., 2000; Akutsu et al., 1999); *Bayesian networks* (Friedman et al., 2000), and *methods based on a steady-state description* (Tegner et al., 2003). Unfortunately, the main drawback of these models is that the gene expression is represented only in the two extreme levels, and therefore all genes are mapped only in a binary state: *on* (1) or *off* (0). This disadvantage makes limited use of such models since the real gene expression levels tend to be continuous rather than binary. An other drawback is also given from their not ability to capture the non-

linear gene regulations, typical feature of the gene regulatory networks. To overcome this limitation, models based on *ordinary differential equations* (Chen et al., 1999) have been designed, which represent a very powerful and flexible model to describe complex relations among more components. One of the most popular and studied approaches, based on ODE, is the *S-system* model, whose structure is rich enough to capture many relevant biological dynamics, and it can models much more complicated GRN behaviour (Wessels et al., 2001). A comprehensive interesting comparative study on the three most used continuous systems based on ordinary differential equations has been made in (Swain et al., 2010) (*S-system*, *artificial neural network* (Vohradsky, 2001), and *general rate law of transcription* (Mendes et al., 2003)), where the advantages and disadvantages of each deterministic model used for modelling gene regulatory networks have been reported. In the last decades, inferring gene regulatory networks from time-series data has attracted a lot of attentions by many researchers in systems biology. It is then important to develop proper models that incorporate a suitable compromise among different requirements, as e.g. computational complexity, the ability to capture nonlinear gene regulations and the ability to handle noisy data. Moreover, it is also able to model much more complicated GRN behaviour (Wessels et al., 2001), and therefore it presents a good compromise between accuracy and mathematical flexibility. The *S-system* model is a type of power-law formalism used to model molecular networks, whose expression rates are described as the difference between the activation and degradation terms of a gene product. It is formally defined as a set of non-linear ordinary differential equations of the form:

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^n X_j^{g_{ij}} - \beta_i \prod_{j=1}^n X_j^{h_{ij}}, \quad (1)$$

where  $n$  is the number of the genes;  $X_i$  is the expression level of the  $i$ -th gene; the exponential parameters  $g_{ij}$  e  $h_{ij}$  represent the effective interaction of  $X_j$  to  $X_i$ . In equation (1), the first term represents all influences that increase  $X_i$ , whilst the second term the ones that decrease  $X_i$ : if  $X_j$  has a positive exponent it means that it has a positive correlation on the aggregation process, whilst if it is negative then the genes are negatively correlated. Of course, if the exponent is zero then there not exist any influence on the aggregation process. From biochemical engineering point of view, the non-negative parameters  $\alpha_i, \beta_i \in [R_l, R_u]$  are called *rate constants*, whereas the real value exponents  $g_{ij}, h_{ij} \in [K_l, K_u]$  are referred to as *kinetic orders*. The aim on the *S-system* model is inferring the set of parameters  $\Omega = \{\alpha_i, \beta_i, g_{ij}, h_{ij}\}$ , such that the fitness function is minimized. Is easy to see how extracting the parameters  $\Omega$  in a genetic network with  $n$  genes is not trivial task due to the high dimensionality of the problem:  $2n(n+1)$  parameters indeed must be inferred. Obviously, the difficulty

of the problem increases by how much information need to be captured. To overcome this limitation and facilitate the regression task a *decoupled variant* of the model has been proposed (Kimura et al., 2005; Noman and Iba, 2006; Vilela et al., 2008a), which reduces the problem in  $n$  sub problems. Through this decomposition strategy, the original optimization problem is divided into  $n$  sub-problems where in any gene  $i$  the parameter values  $(\alpha_i, \beta_i, g_{ij}, h_{ij})$  are individually estimated to attempt to capture the dynamics of the own gene. In this way, the original problem of  $2n(n+1)$  dimensional is reduced in  $n$  sub problems each of  $2(n+1)$  dimension. Thus, in the  $i$ -th sub-problem the expression level of the gene  $i$  is computed by the following ordinary differential equation:

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^N Y_j^{g_{ij}} - \beta_i \prod_{j=1}^N Y_j^{h_{ij}}, \quad (2)$$

where:

$$Y_j = \begin{cases} X_j, & \text{if } i = j \\ \hat{X}_j, & \text{otherwise,} \end{cases}$$

with  $X_j$  computed solving the differential equation (2); and  $\hat{X}_j$  estimated directly from experimentally observed time-series data using differential equation (1). Estimating the inferred set of the parameters is usually evaluated by the *Mean Squared Error* (MSE) between the experimentally observed expression levels, and the ones computed solving the system of equation (1). Therefore, the optimization task is inferring the parameters  $\Omega$  in decoupled form in order to minimize MSE.

### The CMA-ES algorithm

To attempt to inferring the set of parameters of a genetic network using the S-system model, we have adopted the CMA-ES algorithm (Hansen and Ostermeier, 2001), one of the best population-based optimization algorithms that is very suitable primarily on non-linear and non-convex optimization task. Since CMA-ES algorithm is well known inside the evolutionary computation community, in this section we give a short description on its main features. It is a  $(1+1)$  elitist evolutionary strategy that generates candidate solutions by adapting a covariance matrix  $C$ , such that steps promising large fitness progress are sampled more often. Conversely to other self-adaptive evolutionary algorithms, CMA-ES adapts the covariance matrix, at generation  $g$ , by additive updates of the form  $C^{(g)} = \alpha C^{(g-1)} + \beta V^{(g-1)}$ , where  $V^{(g-1)} \in \mathbb{R}^{n \times n}$  is positive definite and  $\alpha, \beta \in R_0^+$  are weighting factors. Let  $v^{(g-1)} \in \mathbb{R}$  a promising mutation step, to increase the probability of sampling  $v^{(g-1)}$  in the next generation, the rank-one update is performed in the equation:  $C^{(g)} = \alpha C^{(g-1)} + \beta v^{(g-1)} v^{(g-1)T}$ . This update strategy shifts the mutation distribution towards the Gaussian with highest probability of generating  $v^{(g-1)}$ . The

CMA-ES algorithm is based on three main procedures: (1) *main loop*, (2) *step size updating procedure*, and (3) *covariance matrix updating strategy*. The CMA-ES main loop follows the classical  $(1+1)$  scheme, where the offspring  $x_{\text{offspring}}$  replaces the parent  $x_{\text{parent}}$  if its fitness value is better. Successively, the algorithm updates the step size, which is based on the heuristic that increases it if the success rate is high, and reducing it otherwise. The procedure performs an update based on a binary variable  $(\lambda_{\text{succ}})$ , which is set to 1 if  $f(x_{\text{offspring}}) \leq f(x_{\text{parent}})$ , with learning parameter  $c_p \in (0, 1]$  using a target success rate  $p_{\text{succ}}^{\text{target}}$ . If  $p_{\text{succ}} > p_{\text{succ}}^{\text{target}}$  the argument is greater then zero and the step increased; if  $p_{\text{succ}} < p_{\text{succ}}^{\text{target}}$ , the argument is smaller than zero and the step size is decreased otherwise it remains unchanged. Finally, the update of the covariance matrix and the evolution path ( $p_c$ ) takes place if  $f(x_{\text{offspring}}) \leq f(x_{\text{parent}})$ , and it depends on the values of  $p_{\text{succ}}$ ; if  $p_{\text{succ}}$  is high the update of  $p_c$  is blocked in order to prevent a fast increase of the  $C$  axis when the step size is low, otherwise the update occurs by an exponential smoothing. The new covariance matrix is a weighted mean of the old matrix and the outer product  $p_c p_c^T$ . Major details on CMA-ES can be found in (Auger and Hansen, 2005; Hansen and Ostermeier, 2001; Cutello et al., 2010).

### Results

For our experiments we have used the classical artificial genetic networks that include an overall of 5 different instances: 2 instances with 2 genes (Vilela et al., 2008a), where 12 parameters need to be inferred for each; 1 instance with 4 genes (Vilela et al., 2008a) and 40 parameters to be inferred; and finally 2 artificial networks with 5 genes (Vilela et al., 2008a; Kikuchi et al., 2003; Noman and Iba, 2006), where 60 parameters must to be inferred. Of course, thanks to these experiments we are also able to evaluate the performances and efficiency of CMA-ES on this new kind of complex optimization task. Due to a limit pages we show in this section the results on the networks with 5 genes. In all experiments, we have considered the ranges where compute parameters  $(\alpha, \beta \in [R_l, R_u])$ , and  $g_{ij}, h_{ij} \in [K_l, K_u]$ , as well as initial conditions, the same ones used in the relative papers from where each instance has been taken into account. About CMA-ES algorithm, instead, we have fixed  $\mu = \lambda = 100$ , and 100 sample points; as termination criterion has been used a maximum number of fitness function evaluation fixed to  $10^8$ . Moreover, each experiment has been performed over 10 independent runs as proposed in (Vilela et al., 2008a). In the first experiments presented in this section we compare CMA-ES with the algorithm proposed in (Vilela et al., 2008a) (in the follows called *Voit's* algorithm), which is based on eigenvector optimization of a matrix formed from multiple regression equations of the linearized decoupled S-system. In these experiments we have tested CMA-ES on artificial gene networks with 2 (two in-

Table 2: CMA-ES versus state-of-the-art optimization algorithms. The comparisons have been done on a genetic network with 5 genes considering the *Euclidean distance* ( $d_{euc}$ ) as evaluation measure. In all algorithms, for each gene has been included the best computed parameters (the rate constants  $\alpha_i$ ,  $\beta_i$  and the kinetic orders  $g_{i,j}$ ,  $h_{i,j}$ ).

gene	$\alpha_i$	$g_{i1}$	$g_{i2}$	$g_{i3}$	$g_{i4}$	$g_{i5}$	$\beta_i$	$h_{i1}$	$h_{i2}$	$h_{i3}$	$h_{i4}$	$h_{i5}$	$d_{euc}$
CMA-ES													
$X_1$	5.0	0.0	0.0	1.0	0.0	-1.0	10.0	2.0	0.0	0.0	0.0	0.0	0.0
$X_2$	10.0	2.0	0.0	0.0	0.0	0.0	10.0	0.0	2.0	0.0	0.0	0.0	
$X_3$	10.0	0.0	-1.0	0.0	0.0	0.0	10.0	0.0	-1.0	2.0	0.0	0.0	
$X_4$	8.0	0.0	0.0	2.0	0.0	-1.0	10.0	0.0	0.0	0.0	2.0	0.0	
$X_5$	10.0	0.0	0.0	0.0	2.0	0.0	10.0	0.0	0.0	0.0	0.0	2.0	
MO-HDE (Liu and Wang, 2008)													
$X_1$	4.95	0.0	0.0	1.007	0.0	-1.011	9.9	1.997	0.0	0.0	0.0	0.0	0.05
$X_2$	9.95	1.992	0.0	0.0	0.0	0.0	9.96	0.0	1.999	0.0	0.0	0.0	
$X_3$	10.22	0.0	-0.968	0.0	-0.002	0.0	10.24	0.0	-0.966	1.998	0.0	0.0	
$X_4$	7.93	0.0	0.0	2.009	0.0	-1.008	9.89	0.0	-0.004	0.0	1.993	0.0	
$X_5$	9.97	0.0	0.0	0.0	1.993	0.0	9.97	0.0	0.0	0.0	0.0	1.996	
coop-CE (Kimura et al., 2005)													
$X_1$	4.917	-0.009	-0.003	1.019	-0.017	-1.014	9.922	2.021	-0.009	0.002	-0.009	-0.009	0.6178
$X_2$	10.03	1.995	0.002	-0.002	0.006	-0.001	10.026	0.002	1.995	-0.002	0.002	0.0	
$X_3$	9.851	-0.005	-0.991	-0.004	-0.003	0.002	9.835	-0.004	-0.993	2.036	-0.01	0.002	
$X_4$	8.02	-0.007	0.006	2.0	-0.002	-0.998	10.054	0.001	0.003	0.008	1.988	0.007	
$X_5$	9.875	-0.002	0.003	0.018	2.015	-0.02	9.892	0.004	0.002	0.008	-0.01	2.017	
HDE (Tsai and Wang, 2005)													
$X_1$	5.0145	0.0	0.0	1.0128	0.0	-1.0031	10.01	1.9936	0.0	0.0	0.0	0.0	0.737
$X_2$	9.9	1.99	0.0	0.0	0.0	0.0	9.871	0.0	1.99	0.0	0.0	0.0	
$X_3$	10.321	0.0	-0.963	0.0	0.0	0.0	10.344	0.0	-0.9594	1.9987	0.0	0.0	
$X_4$	7.99	0.0	0.0	2.0157	0.0	-1.0026	9.981	0.0	0.0	0.0	2.0018	0.0	
$X_5$	9.966	0.0	0.0	0.0	1.985	0.0	9.967	0.0	0.0	0.0	0.0	1.997	
TDE <sub>1</sub> (Noman and Iba, 2005)													
$X_1$	4.762	-0.021	-0.021	0.993	0.0	-1.013	9.607	1.916	0.0	0.0	0.0	0.0	2.0597
$X_2$	10.08	1.99	-0.001	0.035	0.0	0.0	9.817	0.0	1.938	0.0	0.012	0.0	
$X_3$	9.823	0.0	-1.00	-0.008	0.0	-0.001	9.835	0.0	-1.00	2.031	0.0	0.0	
$X_4$	7.182	0.0	-0.036	2.039	-0.052	-1.044	9.415	0.0	0.0	0.0	2.034	0.0	
$X_5$	10.103	0.0	0.005	0.05	1.997	-0.003	10.049	0.0	0.0	0.0	0.0	2.005	
TDE <sub>2</sub> (Noman and Iba, 2006)													
$X_1$	4.99	0.0	-0.008	0.98	-0.004	-0.997	10.003	1.978	0.0	0.0	0.0	0.0	2.2774
$X_2$	10.051	1.995	0.004	0.009	0.002	-0.002	10.06	0.0	1.998	0.012	0.0	0.01	
$X_3$	9.936	0.004	-1.001	-0.001	0.0	0.0	9.937	-0.004	-1.001	2.007	0.0	0.001	
$X_4$	8.032	0.0	-0.011	1.949	0.0	-0.996	10.153	0.0	0.007	0.0	1.972	0.0	
$X_5$	10.011	0.0	0.003	0.023	2.002	-0.009	9.992	0.006	0.0	0.002	0.0	1.99	
PEACE1 (Kikuchi et al., 2003)													
$X_1$	5.9	0.0	0.0	0.9	0.0	-0.9	10.6	1.7	0.0	0.0	0.0	0.0	74.0434
$X_2$	10.0	2.1	0.0	0.0	0.0	0.0	10.2	0.0	2.1	0.0	0.0	0.0	
$X_3$	9.6	0.0	-0.9	0.0	0.0	0.0	9.7	0.0	-0.9	2.3	0.0	0.0	
$X_4$	9.4	0.0	0.0	1.9	0.0	-0.9	11.5	0.0	0.0	0.0	1.8	0.0	
$X_5$	10.2	0.0	0.0	0.0	2.1	0.0	10.2	0.0	0.0	0.7	0.0	1.9	

Table 1: Success rate (SR) obtained by CMA-ES and *Voit's* algorithm (Vilela et al., 2008a) on 10 independent runs. Both algorithms have been tested on an artificial network with 5 genes.

gene	CMA-ES	<i>Voit's</i> alg. (Vilela et al., 2008a)
$X_1$	100%	100%
$X_2$	100%	100%
$X_3$	<b>100%</b>	0%
$X_4$	100%	100%
$X_5$	100%	100%

stances – normal and rescaled), 4 (rescaled) and 5 components. For each instance three different data sets have been used. All details about the instances can be found in the relative additional material (Vilela et al., 2008b). On the

artificial genetic network with 2 genes both algorithms are comparable in term of success rate (*SR*), that is how many times the algorithm infers the parameters target. However, if we compare the results where both algorithms fail, it is possible to see as CMA-ES outperforms the compared algorithm in terms of *Euclidean distance* between the computed parameters and estimated parameters. This means that CMA-ES is able inferring the set of parameters closer to the estimated ones. The rescaled 2 genes network is, instead, equal to the normal one where however  $\alpha$  and  $\beta$  are multiplied by a constant. In this experiment, using all three data sets, CMA-ES has been found the  $\Omega$  target for each gene in all 10 runs with  $SR = 100\%$ , except for the gene  $X_2$  of the 3rd data set where  $SR = 70\%$ . The compared algorithm, instead, presents  $SR = 100\%$ , excepts for the gene  $X_2$  in the 2nd and 3rd data set, with respectively  $SR = 80\%$  and  $SR = 70\%$ . In the overall, we can say that CMA-ES outperforms the *Voit's* algorithm (Vilela et al., 2008a) in both artificial genetic networks with 2 components (in terms of

success rate and Euclidean distance). For the network instance with 4 genes, both algorithms are equivalent, since they have been able to find always the network target in all 10 runs ( $SR = 100\%$ ). About the experiments on the artificial network with 5 genes, CMA-ES and the *Voit's* algorithm have been compared on three different data sets. However, due to the limit pages we report in Table 1 only the results obtained on the 2nd data set. Inspecting the Table, is possible to see that albeit both algorithms reach a  $SR = 100\%$  for the genes  $X_1$ ,  $X_2$ ,  $X_4$ , and  $X_5$  CMA-ES is able also to inferring the estimated parameters for the gene  $X_3$  on all 10 runs ( $SR = 100\%$ ), where instead *Voit's* algorithm fails with a zero success rate. Fig. 1 shows the gene expression

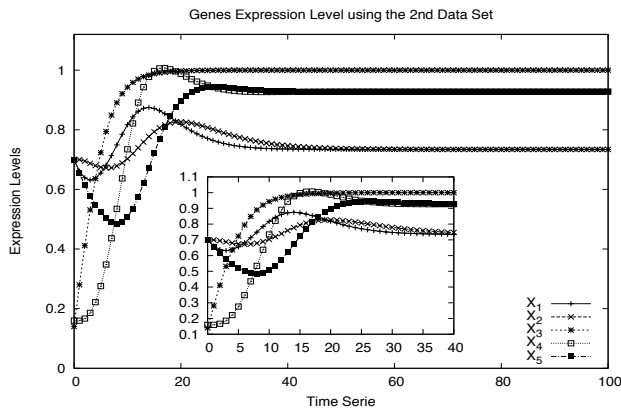


Figure 1: Gene expression levels computed by CMA-ES on the artificial network with 5 genes using 100 sample points.

levels computed by CMA-ES on the second data set. Looking the Table 1 is easy to understand how these curves represent exactly the gene expression levels of the target network. These plots have been obtained with a time-series based on 100 sample points. Instead, about the experiments in order to the other two data sets, we can say that from the obtained results is clear as both algorithms are able to estimate for all genes the target parameters, with  $SR = 100\%$ , about the first data set, unlike the third data set, where CMA-ES shows best performances (in the overall) than the *Voit's* algorithm. In this last data set, CMA-ES outperforms *Voit's* Algorithm on the genes  $X_1$ ,  $X_4$  and  $X_5$  with a success rate of 100%, with respect to a success rate of 30% ( $X_1$ ), and 0% ( $X_4$  and  $X_5$ ). Only on the gene  $X_3$  CMA-ES fails over all 10 runs unlike the *Voit's* algorithm that produces a  $SR = 70\%$ . However, although CMA-ES seems not comparable to the *Voit's* algorithm for the gene  $X_3$ , if we take into account only the remaining 30% of the computed parameters by *Voit's* algorithm, where  $SR = 0\%$ , and we compare them with all ones produced by CMA-ES is possible to note how the inferred parameters by our algorithm seem better in term of Euclidean distance from the expected target parameters. To better evaluate the robustness of our proposed

algorithm on these kinds of complex optimization tasks we have compared CMA-ES with state-of-the-art algorithms on S-system models. For these new experiments the *Euclidean distance* from the estimated parameters has been chosen as evaluation measure. A new instance with 5 genes has been considered that is different from the previous one because different ranges have been used where compute  $\Omega$  parameters. For this instance, moreover, it is important to point out that CMA-ES has been tested on 100 independent runs, producing an high success rate very closer to 100%. In Table 2 we report the comparisons of CMA-ES with the state-of-the-art, where only the best results for all algorithms have been included. The algorithms compared with CMA-ES are: (1) MO-HDE (Liu and Wang, 2008), a multi-objective optimization approach based on an hybrid differential evolution; (2) coop-CE (Kimura et al., 2005), a cooperative Co-evolutionary algorithm; (3) HDE (Tsai and Wang, 2005), a hybrid differential evolution; (4) and (5) two different versions of trigonometric differential evolution ( $TDE_1$  (Noman and Iba, 2005) and  $TDE_2$  (Noman and Iba, 2006)); and finally (6) PEACE1 (Kikuchi et al., 2003) based on a Genetic algorithm. From the Table is clear as CMA-ES produces the best performances with zero Euclidean distance, whilst the best among the compared algorithms was able to reach 0.05 as Euclidean distance from the estimated parameters. The genetic algorithm is instead the one with worst performances. It is possible to claim that CMA-ES outperforms the current state-of-the-art optimization algorithms on S-system models.

### Time-To-Target Analysis

Time-To-Target plots (Aiex et al., 2002) are a method to characterize the running time of stochastic algorithms to solve a given computational optimization problem. They display the probability that a given algorithm will find a solution as good as a target within a given running time. Nowadays they are standard graphical methodologies for data analysis to compare the empirical and theoretical distributions (Aiex et al., 2002, 2007). By Time-To-Target analysis two kinds of plots are produced: *QQ*-plot with superimposed variability information, and superimposed empirical and theoretical distributions.

We ran CMA-ES on the genetic network with 5 genes, and where the success rate in inferring the set of parameters of all genes is 100%. For this kind of experiments a different termination criterion has been properly tuned: until finding the parameters target for each gene ( $SR = 100\%$ ). Because larger is the number of runs closer is the empirical distribution to the theoretical distribution, the plots presented in this section have been produced after 100 independent runs. The Fig. 2 shows the convergence process produced by CMA-ES using `tttplots.pl` on  $n = 5$  network instance. In the top plot is showed the comparisons among empirical and theoretical distributions, whilst in bottom one is showed the



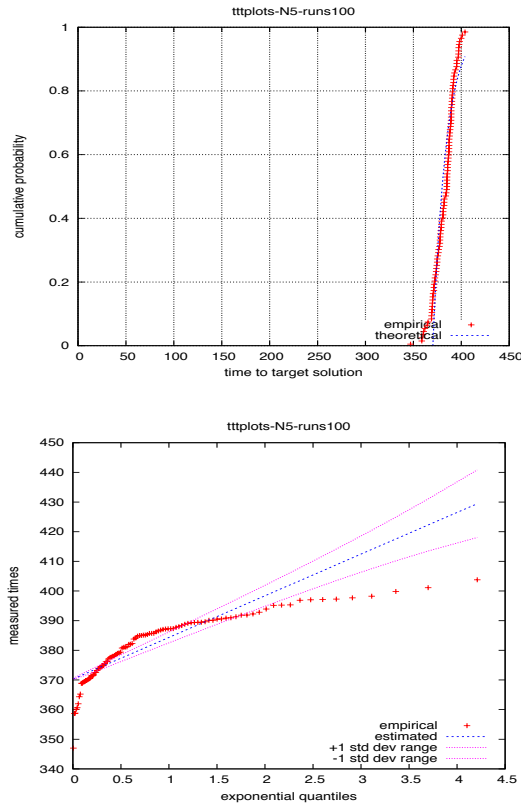


Figure 2: Time-To-Target plots for the network with 5 genes.

*QQ*-plot with variability information. Looking the top plot is possible to see as the curves of the empirical and theoretical distribution are equivalent. A different behaviour instead is showed on the bottom plot; this is likely due, since CMA-ES finds in quick way the parameters target.

### Sensitivity Analysis – The Morris Method

The *Morris method* is one of the most popular models to evaluate the importance of any single parameter of a given system, showing own the main interactions between the parameters. In this method, any parameter assumes a discrete number of values chosen inside a range of variation; these values are called *levels*. Morris (Morris, 1991) has used a sensitivity analysis based on the *elementary effect* of the  $j$ -th parameter, defined as:

$$EE_j(p^*) = \frac{[f(p_1^*, \dots, p_{j-1}^*, p_j^* + \Delta, p_{j+1}^*, \dots, p_{N_p}^*) - f(p^*)]}{\Delta},$$

where  $\Delta$  is a predetermined multiple of  $1/(k-1)$  ( $k$  is the number of levels). To understand what are the parameters, which influence on the output we have performed the Morris method for the S-Systems models, with  $n = (2, 4, 5)$  genes. The obtained results are showed in Fig. 3. Two sensitivity measures,  $\mu_j$  and  $\sigma_j$ , have been evaluated for any parameter

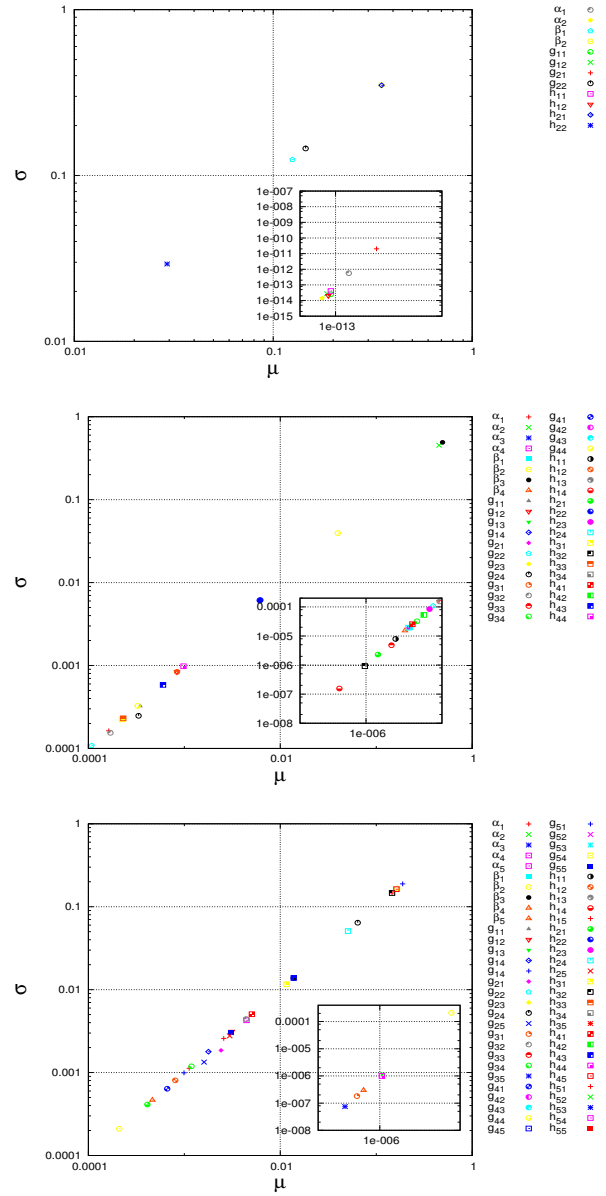


Figure 3: Sensitivity analysis by Morris method. A high value of  $\mu$  indicates a parameter with an important overall influence on the output. A high value of  $\sigma$  indicates a parameter involved in interaction with other parameters or whose effect is nonlinear. The results show high  $\mu$  and  $\sigma$  values for  $h_{21}$  and  $\beta_2$  for the GRN with N=2 genes (top plot), for  $\alpha_2$  and  $\beta_3$  for the GRN with N=4 (middle plot), for  $g_{24}$ ,  $g_{44}$ ,  $g_{51}$ ,  $h_{15}$ ,  $h_{24}$ ,  $h_{32}$ ,  $h_{45}$  for the GRN with N=5 (bottom plot).

$j$ : the first represents an estimate of the mean of the distribution of the elementary effects, whilst the last indicates its standard deviation. High value of mean represents an important overall influence on the output by the given parameter; whereas high values of the standard deviation for the  $j$ -th parameter means that it is involved in interaction with other

parameters, or whose effect is nonlinear.

## Conclusion

In this research paper we have presented a new approach for inferring the parameters in S-system models of gene regulatory networks. The CMA-ES algorithm has been used for this complex optimization task, and 5 different instances have been taken into account to evaluate its performances and its robustness: 2 instances with 2 genes - 12 parameters need to be inferred for each network; 1 instance with 4 genes - 40 parameters to be inferred; and 2 artificial networks with 5 genes - 60 parameters must to be inferred for each instance. The proposed algorithm has been compared with 7 state-of-the-art algorithms: (1) *Voit's* algorithm, (2) MO-HDE; (3) coop-CE; (4) HDE; (5) and (6) two different versions of trigonometric differential evolution; and (7) PEACE1. The first experiments have been done on the instances with  $n = (2, 4, 5)$  components taken from (Vilela et al., 2008a), comparing also CMA-ES with *Voit's* algorithm. Due to the limit pages only the Table with the results obtained on a genetic network with  $n = 5$  genes has been included in the paper. Analyzing the results obtained on all instances appear to be clear how CMA-ES is more able to estimating in a better way the parameters target, either in order to the success rate and in term of Euclidean distance. To have a better knowledge about the robustness of CMA-ES, we have compared it also with the current state-of-the-art algorithms, where the Euclidean distance has been used as evaluation metric. From these comparisons, CMA-ES is the only algorithm able to inferring the parameters effectively. Reviewing all experiments from an overall point of view is possible to claim that CMA-ES is an effective optimization algorithm for complex tasks, ranking as among one of the best reverse engineering methodologies on S-system models. Finally, in this research paper has been also included a study on the convergence process of CMA-ES through Time-To-Target plots, which are a way to characterize the running time of stochastic algorithms; and a global sensitivity analysis method, the Morris' algorithm.

## References

- Aiex, R. M., Resende, M. G. C., and Ribeiro, C. C. (2002). Probability distribution of solution time in grasp: An experimental investigation. *Journal of Heuristics*, 8:343–373.
- Aiex, R. M., Resende, M. G. C., and Ribeiro, C. C. (2007). Ttplots: A perl program to create time-to-target plots. *Optimization Letters*, 1:355–366.
- Akutsu, T., Miyano, S., and Kuhara, S. (1999). Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In *Proceedings of Pacific Symposium on Biocomputing*, pages 17–28.
- Auger, A. and Hansen, N. (2005). A restart cma evolution strategy with increasing population size. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 1769–1776. IEEE Press.
- Barrett, C. L., Kim, T. Y., Kim, H. U., Palsson, B., and Lee, S. Y. (2006). Systems biology as a foundation for genome-scale synthetic biology. *Current Opinions in Biotechnology*, 17(5):488–492.
- Chen, T., He, H. L., and Church, G. M. (1999). Modeling gene expression with differential equations. In *Proceedings of Pacific Symposium on Biocomputing*, pages 29–40.
- Cho, K. H., Choo, S. M., Jung, S. H., Kim, J. R., Choi, H. S., and Kim, J. (2007). Reverse engineering of gene regulatory networks. *IET System Biology*, 1:149–163.
- Csete, M. and Doyle, J. (2002). Reverse engineering of biological complexity. *Science*, 295:1664–1669.
- Cutello, V., Nicosia, G., Pavone, M., and Stracquadanio, G. (2010). Entropic divergence for population based optimization algorithms. In *Proceedings of IEEE World Congress on Computational Intelligence*, pages 1–8. IEEE Press.
- Davidson, E. and Levin, M. (2005). Gene regulatory networks. In *Proceedings of the National Academy of Sciences of the United States of America - PNAS*, volume 102(14), page 4935.
- D'haeseleer, P., Liang, S., and Somogyi, R. (2000). Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 18(8):707–726.
- Friedman, N., Linial, M., Nachman, I., and Peer, D. (2000). Using bayesian networks to analyze expression data. In *Proceedings of Research in Computational Molecular Biology - RECOMB*, pages 127–135.
- Hansen, N. and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195.
- Irvine, D. H. and Savageau, M. A. (1990). Efficient solution of nonlinear ordinary differential equations expressed in s-systems canonical form. *SIAM journal of Numerical Analysis*, 27(3):704–735.
- Kabir, M., Noman, N., and Iba, H. (2010). Reverse engineering gene regulatory network from microarray data using linear time-variant model. *BMC Bioinformatics*, 11 (suppl. 1) S56.
- Kikuchi, S., Tominaga, D., Arita, M., Takahashi, K., and Tomita, M. (2003). Dynamic modeling of genetic networks using genetic algorithm and s-system. *Bioinformatics*, 19(5):643–650.

- Kimura, S., Ide, K., Kashihara, A., Kano, M., Hatakeyama, M., Masui, R., Nakagawa, N., Yokoyama, S., Kuramitsu, S., and Konagaya, A. (2005). Inference of s-system models of genetic networks using a cooperative algorithm. *Bioinformatics*, 21(7):1154–1163.
- Lee, D. (2005). Component-based software architecture for biosystem reverse engineering. *Biotechnology and Bio-process Engineering*, 10:400–407.
- Liu, P. K. and Wang, F. S. (2008). Inference of biochemical network models in s-system using multiobjective optimization approach. *Bioinformatics*, 24(8):1085–1092.
- Mendes, P., Sha, W., and Ye, K. (2003). Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics*, 19:122–129.
- Morris, M. D. (1991). Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2):161–174.
- Noman, N. and Iba, H. (2005). Reverse engineering genetic networks using evolutionary computation. *Genome Informatics*, 16(2):205–214.
- Noman, N. and Iba, H. (2006). Inference of genetic networks using s-system information criteria for model selection. In *Proceedings of ACM Genetic and Evolutionary Computation Conference*, pages 263–270. ACM.
- Noman, N. and Iba, H. (2007). Inferring gene regulatory networks using differential evolution with local search heuristics. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(4):634–647.
- Schlitt, T. and Brazma, A. (2007). Current approaches to gene regulatory network modelling. *BMC Bioinformatics*, 8.
- Sirbu, A., Ruskin, H. J., and Crane, M. (2010). Comparison of evolutionary algorithms in gene regulatory network model inference. *BMC Bioinformatics*, 11(58).
- Swain, M. T., Mandel, J. J., and Dubitzky, W. (2010). Comparative study of three commonly used continuous deterministic methods for modeling gene regulation networks. *BMC Bioinformatics*, 11(459).
- Tegner, J., Yeung, M. K. S., Hasty, J., and Collins, J. J. (2003). Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. In *Proceedings of the National Academy of Sciences of the United States of America - PNAS*, volume 100(10), pages 5944–5949.
- Tsai, K. Y. and Wang, F. S. (2005). Evolutionary optimization with data collocation for reverse engineering of biological networks. *Bioinformatics*, 21(7):1180–1188.
- Vilela, M., Chou, C., Vinga, S., Vasconcelos, A., Voit, E. O., and Almeida, J. (2008a). Parameter optimization in s-system models. *BMC Systems Biology*, 2(35).
- Vilela, M., Chou, C., Vinga, S., Vasconcelos, A., Voit, E. O., and Almeida, J. (2008b). Supplementary material to *parameter optimization in s-system models*.
- Vohradsky, J. (2001). Neural network model of gene expression. *The FASEB journal: official publication of the Federation of American Societies for Experimental Biology*, 15(3):846–854.
- Wessels, L. F. A., Van Someren, E. P., and Reinders, M. J. T. (2001). A comparison of genetic network models. In *Pacific Symposium on Biocomputing*, volume 6, pages 508–519.

# The origins of communication revisited

Jordi Arranz<sup>1</sup>, Jason Noble<sup>1</sup> and Eric Silverman<sup>1</sup>

Institute for Complex Systems Simulation, University of Southampton  
jordi.arranz@soton.ac.uk

## Abstract

Quinn (2001) sought to demonstrate that communication between simulated agents could be evolved without pre-defined communication channels. Quinn's work was exciting because it showed the potential for ALife models to look at the real origin of communication; however, the work has never been replicated. In order to test the generality of Quinn's result we use a similar task but a completely different agent architecture. We find that qualitatively similar behaviours emerge, but it is not clear whether they are genuinely communicative. We extend Quinn's work by adding perceptual noise and internal state to the agents in order to promote ritualization of the nascent signal. Results were inconclusive; philosophical implications are discussed.

## Introduction

Artificial life researchers have been modelling the evolution of communication for some time now (for early examples see MacLennan, 1992; Werner and Dyer, 1992; Noble and Cliff, 1996). Communication is of interest in our field for a range of overlapping reasons, most notably because it is associated with two of the major transitions in evolution (Maynard Smith and Szathmáry, 1995): the jump from solitary to social living; and the later development of language and culture in our own species. ALife's agent-based simulations are a natural match for this research area as they can provide emergent explanations of communication and related co-evolutionary phenomena that are not possible using more traditional modelling techniques.

However, prior to the publication of a seminal paper by Quinn (2001), computational models of the origins of communication and language were missing an important opportunity. Influenced by game theory, by the long shadow of Shannon and Weaver (1949), and by what Lakoff and Johnson (1980) called "the conduit metaphor" for communication, modellers tended to assume that a signalling channel already existed between the relevant agents, and that the thing to be explained was how and why that signalling channel would come to be used for honest, coherent, and reliable communication. MacLennan's (1992) early work, for example, imagined agents with eight possible world states, each

matched with one of eight preferred responses, and a convenient library of eight ready-made symbols that had to be mapped, over evolutionary time, in a way that would allow pairs of agents to communicate and thus perform optimally.

These kinds of models ignored the apparently vicious circle involved in the evolution of natural communication systems: for a signal to have any meaning, for it to be worth producing, there has to be a community of responders. But why would the appropriate response behaviour already exist if the signal itself has not evolved yet?

This paradox had been noted, and resolved, many years earlier by the ethologists (Tinbergen, 1964). The two key concepts in the ethological picture of the evolution of communication are "intention movements" — non-signals which provide the raw materials for signal evolution — and the subsequent "ritualization" of the nascent signal. Intention movements have not been selected for *per se*; they are simply a physically necessary step in performing some action, e.g., an animal that intends to bite an opponent *must* bare its teeth before doing so. Intention movements thus provide information about future behaviour, and it is not difficult to see how such movements, coupled with the complementary ability to recognize them, might provide the seeds for the evolution of a communication system. Ritualization is what happens when an initially irrelevant movement such as teeth-baring starts to be of informational value to other animals. The ethologists, assuming that the reliable transmission of information would always carry a selective advantage, thought that the original cue would then be exaggerated or stylized in the interests of reducing ambiguity.

Inspired in part by the ethological perspective, Quinn (2001) sought to demonstrate in a simulation that communication between agents could be evolved without pre-defined communication channels; in other words, he hoped to produce a genuine account of the *origin* of communication. Quinn's point was that by supplying a signalling channel and a library of signals, most of the previous models were assuming the existence of exactly what it was they should be trying to explain. He began with pairs of agents that were linked only by basic sensory-motor interaction, i.e., if one

agent moved this could be detected by the visual system of the other agent. The agents were then faced with an explicitly cooperative task: moving their joint centre of mass as far as possible within a time limit. A genetic algorithm (GA) was used to select for agents that, when paired with another member of the population, managed to coordinate their behaviour and score highly on the task. Quinn interpreted his results as showing that communication had evolved in the form of a dance-like negotiation process between the agents that was followed by matched movement away from their starting positions. Note that no explicit role allocation had been forced on the agents: each one was equally likely to end up as the leader or the follower in the movement phase.

Quinn's work was exciting because it showed the potential for ALife models to look at the real origin of communication, rather than just the conditions under which it could be maintained in a system where it was already possible. The model is appealing in that it provides a great example of the kind of emergent explanation that ALife can provide, and a potential bridging account between two levels of description (i.e., the level of raw sensory-motor interaction and the level of symbols and reference). It is also a valuable contribution to the biological literature on communication because it lends support to the ethological theory of intention movements and ritualization. Finally, Quinn (2001) is a very popular paper, having been cited 113 times as of April 2011, according to Google Scholar.

However, Quinn's work has never been replicated. We feel that precisely because Quinn's approach is so promising, it is important to establish its generality before going further: one goal of the current paper is to check whether Quinn's central result is robust. Quinn was working in the area of evolutionary robotics and used a fairly detailed model of a real robot; he also employed a continuous-time recurrent neural network (CTRNN) as the evolvable control architecture. What if his result was a freak occurrence, and turned out to be contingent on some detail of the robot's sensory system or cognitive architecture? The general finding should be robust across these specific details if it is going to be of any value, and therefore we have attempted to replicate Quinn's work using a different model of agent perception and movement, as well as a different evolvable control architecture.

We also want to ask: did Quinn pick the right task? He showed the emergence of (at least) a coordination protocol between pairs of agents, but did he definitively show the evolution of communication? This in turn raises questions about how to define communication and how to distinguish it from "mere" coordinated behaviour; we will address these issues below. Scheutz and Schermerhorn (2008) make the point that in many simple ALife scenarios, there may not in fact be any selective pressure for a communicative solution, and we feel this may regrettably apply in the Quinn case.

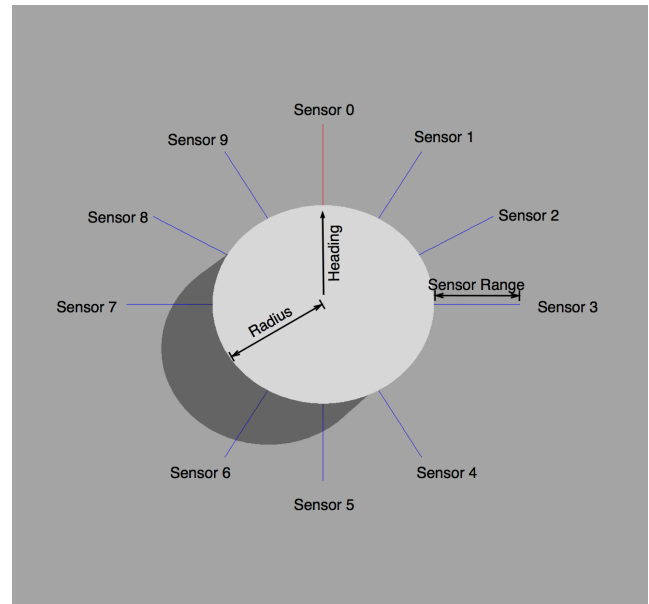


Figure 1: The layout of the ray-cast sensors of our agents. Note that this is not an exact replication of Quinn's simulated Khepera robots. The diagram is not to scale: robot diameter is 55 mm and maximum sensor range is 50 mm.

## The model

Our goal in the first instance is to find out how general Quinn's result was, and thus we have set up a similar task but used a completely different agent architecture. Quinn's agents were fairly realistic simulations of a *Khepera* — these are small, low-cost cylindrical robots, 55 mm in diameter and 30 mm in height, with two independent motors driving two wheels, and a set of eight infra-red (IR) proximity sensors giving the robot the ability to perceive nearby objects. We constructed our own 2D simulator that was less detailed than Quinn's. Our agents are of the same size and shape as a *Khepera* robot but the sensors are of a different kind, number, and position: see figure 1 for details. Most of the changes we have made to Quinn's design are arbitrary, and that is exactly the point. We need to keep certain basic features the same so that the coordinated movement task is both recognizable and feasible, but beyond that our simulation will work best as a measure of the generality of Quinn's result if it is as different as possible.

The agents have been simplified in several ways, but the cylindrical shape has been kept in order to make them rotationally invariant and thus prevent any simple short-cuts that would allow one agent to detect the orientation of its partner. The drive wheels of the *Khepera*, and details such as inertia and friction, are no longer simulated. Movement and rotation are simply transforms in the two-dimensional simulated environment; agents are moved and rotated around their centre-point. The eight IR sensors have been replaced

by ten ray-cast sensors. They operate by throwing a ray of a certain length and infinitesimal width along the vector the sensor is pointing to. If the ray collides with the other agent (there is nothing else in the environment) within its 50 mm range, the sensor reports the collision distance. If the ray gets to the end of its range without colliding with anything (either because the other agent is not present in that direction, or is more than 50 mm away) the sensor reports 51 mm. Since the amount of space covered by these sensors is significantly smaller than the overlapping fan-shaped response areas of the IR sensors, two additional sensors have been added, bringing the total to ten per agent.

Instead of using a CTRNN as a controller, as Quinn did, our model is based on a simple production-rule system. This is much like a classifier system but with the real-time learning capability removed. Every rule or classifier is composed of a set of ten sensor threshold values, logical operators linking each of these, a comparator condition describing how the sensor values should be compared with the thresholds (less than, greater than, or equal to) and an associated behaviour. Classifiers are fired when the sensory input of the agent at a given time-step matches the classifier condition. When no classifier can be matched a default behaviour is chosen. Every classifier also has a “weight” to avoid clashes when more than one classifier matches the sensory input. In such cases the highest-weighted classifier is fired. Note that the weight of a classifier is not altered by experience: it is a purely random value which can be affected by mutation as can the rest of the classifier.

Agents with internal state are introduced later on in the paper (the initial agents do not have internal state) and they are effectively finite-state machines. Classifiers are specific to a particular internal state of the agent, and when a classifier is fired the state of the agent changes to the output state of the classifier. If no classifier can be matched, there is a default output state, and thus any time the default behaviour is fired, the agent switches to this default state.

In order to make this a replication, the task the agents face is exactly the same as in Quinn’s work: a pair of agents must move their joint centre of mass as far as possible while staying within each other’s sensor range and without colliding. We used much the same type of GA as Quinn did to evolve the population of agents, but some of the parameters employed, as well as the way fitness is computed, are different. As in the original model, there is no predefined role allocation. Agents are drawn randomly from a population of 25 individuals and evaluated in pairs. The initial positions and angles of every pair are not randomly generated but picked from a predefined set (see Quinn’s original paper for more details). Each pair is given 15s (in simulation time) to solve the task. Evaluation is performed in discrete time steps of 0.25s; at every time step new sensor values are computed for both agents; and finally the agent behaves according to its sensory input through the activation of the

highest-weighted matching classifier. Each agent in the pair gets the same score depending on their joint performance. A selection process keeps the best 60% of agents and deletes the rest in every generation, with new agents being created through recombination and mutation of the successful individuals of the previous generation. Recombination is performed as an uniform macro crossover operation by combining the classifiers from both parents. Since recombination is performed at a macro scale, classifiers are never split. Every classifier element is susceptible of mutation. Mutation rates are, depending on the number of state bits, 0.21 (stateless case), 0.22 (1 bit state case) or 0.23 (2 bits state case).

The fitness of every pair of agents is computed as an average over two different terms. The first term measures whether or not the agents are in each other’s sensory range and is itself averaged across all simulation time steps. This term is important in shaping effective solutions, as the agents are effectively very short-sighted and moving out of sensor range is usually a disaster for the over-arching goal of moving the joint centre of mass in a consistent direction. The score of an agent on a given time step is computed as an exponential decay function on the distance to the other agent. If an agent is in sensor range the fitness obtained is 1.0, otherwise fitness decreases exponentially with the distance. The maximum distance is computed as the maximum linear distance an agent could achieve given its linear velocity and the overall simulation time.

At the end of the simulation the second fitness term is computed: it measures the distance that the agents have travelled. If *either* agent has travelled at least 250 mm then this component of fitness is 1.0. If the agents have travelled a shorter distance from their starting positions, this fitness component will be the quotient of the distance travelled by whichever agent has travelled the furthest, over the target distance. Note that an agent could travel approximately 500 mm — double the target distance — during the time available if it moved away in a straight line, which means that the fitness function allows the agents a reasonable amount of time for potential communication before movement begins in earnest.

Even though the overall goal is moving the joint centre of mass, we do not measure this directly. Optimal performance is achieved by staying in sensor range *and* moving as far as possible. The final fitness score is the average of the two terms described above, and thus the maximum score is 1.0. Fitness scores of 0.5 are relatively easily achieved by either not moving at all (thus staying in sensor range and scoring highly on the first component) or moving off in random directions at full speed (scoring highly on the second component).

Finally, at the end of a generation the final fitness of each agent is equal to the average of its scores across many different evaluations with different partners and in different initial positions. Note that all initial positions have the agents start-

	<i>Front</i>				<i>Right</i>				<i>Rear</i>				<i>Left</i>							
	S0		S1		S2		S3		S4		S5		S6		S7		S8		S9	Behaviour
Rule 1	49	∨	9	∧	35	∧	37	∧	29	∨	13	∧	33	∨	29	∧	18	∧	19	Forward
Rule 2	14	∨	16	∧	47	∧	37	∧	47	∧	20	∧	38	∨	10	∧	23	∨	26	Backward
Rule 3	23	∧	39	∨	49	∨	14	∧	40	∨	16	∨	35	∨	8	∧	27	∧	20	Rotate CCW
Rule 4	49	∨	9	∧	2	∧	37	∧	29	∧	19	∧	4	∨	37	∧	26	∨	19	Forward
Rule 5	3	∧	9	∧	15	∨	43	∧	14	∧	8	∧	6	∨	44	∧	38	∨	20	Backward
Default																				Rotate CCW

Table 1: The production rule set of a high-performing strategy found during the replication runs (this rule set leads to perfect performance on the task). For each rule, the table lists the threshold value in mm for each of the ten sensors, and the logical operator (either “and” or “or”) used to link them. In every case the rules used the “less than” comparator, i.e., the value would be true if the other agent was detected at the given distance or closer. Note the mix of “forward”, “backward”, and “rotate counter-clockwise” behaviours that combine to produce coordinated movement. An agent that could detect nothing within sensor range would fall through to the default behaviour of counter-clockwise rotation.

ing inside each other’s 50 mm sensor range.

### Replication results

We ran our simulation 30 times, with each run lasting 2000 generations.

Quantitatively our mean and maximum fitness values were similar to Quinn’s despite the differences in the agent architecture, the GA, and the fitness function. Some of the agents scored very high and even perfect fitness levels although these could not be maintained in the long run as mutation pressure prevented the population as a whole from adopting an optimal strategy. Table 1 shows one of the best rule sets evolved.

Qualitatively, we have analyzed in detail the kinds of strategies that evolved in the most successful runs. Although many different strategies evolved that could accomplish the coordination task, we found that the most common and the most successful one we observed fits reasonably well with the main strategy described by Quinn. Figure 2 illustrates the sequence of behaviours. Both agents start rotating counter-clockwise (A) until the first agent (shown in brown) reaches its favoured alignment relative to the second agent (shown in white) and starts moving one step forward and one step backwards in order to “signal” its readiness and direction to the second agent (B). In the meantime, the second agent keeps rotating counter-clockwise until it matches the first agent’s alignment (C). When both agents are aligned and pointing in opposite directions, the first agent starts moving backwards while the second agent starts moving forward, and thus they move together until the end of the time frame (D). Many variations on this strategy exist, with varying degrees of speed and reliability in achieving alignment. Quinn also notes that the strategy he picked to illustrate the behaviour of the agents is just one of the simplest cases among many variants observed.

The change in the number of collisions over evolutionary time also matches Quinn’s results. The collision rate is ex-

tremely high in the early generations but rapidly decreases as fitness increases. Sudden decreases in the collision rate usually match fitness jumps even though our implementation does not include an explicit penalty for collisions. We can also confirm that, as Quinn stated, the evolution of successful behaviours is extremely sensitive to the initial conditions used (the starting distance between the two agents and their relative orientations) as well as to how the agents are evaluated. In essence every agent has to be evaluated with every possible angle and distance: random runs in which every agent is evaluated with different randomly chosen starting distances and relative orientation angles are completely unsuccessful.

There were many differences introduced between Quinn’s setup and our own, notably the use of a different sensory system and control architecture. Nevertheless we managed to replicate Quinn’s findings: very similar behaviours evolved. We therefore suggest that the emergence of coordinated (and possibly communicative) behaviour to solve this type of task is likely to be a general and framework-independent finding.

### Re-examination of the Quinn paradigm

In the previous section we reported the successful replication of a dance-like negotiation phase between the pairs of agents. This is a pleasing result as it goes some way towards showing that Quinn’s findings are general. However, we did not observe any unequivocal “ritualization” process by which the signal became more exaggerated over time. This led us to wonder whether our agents were really communicating at all.

So what do we mean by communication anyway? Should we expect a sharp dividing line between coordinated behaviour and “true” communication? Some ideas from the philosopher Millikan (1984) will be useful here. She argues that although there is no *sharp* line between those two categories, there is certainly a distinction worth making. Millikan lays out four classes of representational phenomena,

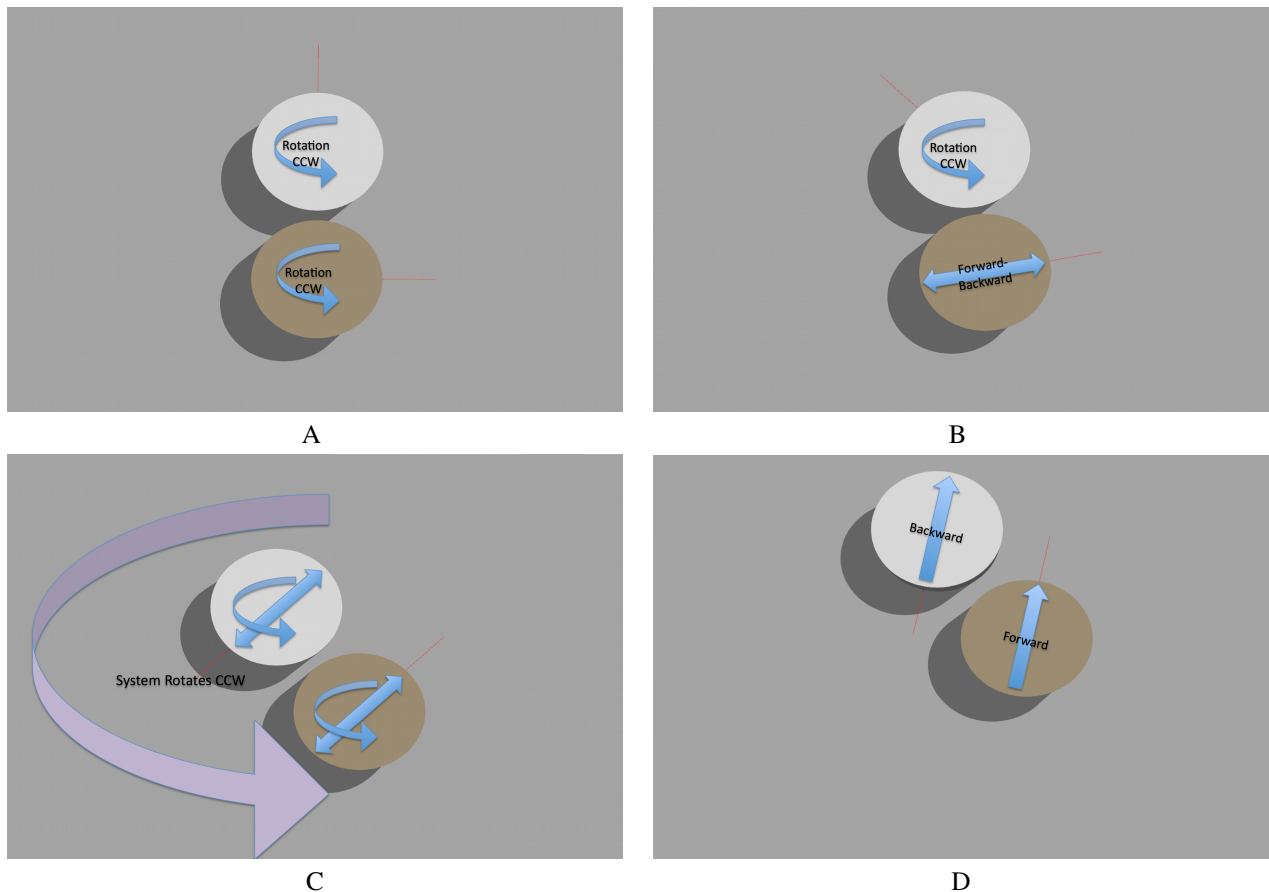


Figure 2: Illustration of the evolved sequence of behaviours in a typical case. A: agents rotate until one reaches a favoured orientation. B: the first agent to achieve this starts moving backwards and forwards. C: the second agent orbits the first until it is aligned in the opposite direction. D: the two agents move away together, with the second agent moving in reverse.

in order of increasing sophistication: tacit suppositions, intentional icons, inner representations, and mental sentences. The first two are all we will need given the simplicity of our agents. (Millikan's typology was initially directed at the issue of what might count as an internal representation within a single organism but it is relevant to our purposes as she sees communication as simply the exchange of representations between organisms.)

Tacit suppositions occur when the design of an organism meshes so neatly with a feature of the environment that it is tempting to say the design "represents" that feature. For example, if a biological clock produces a cycle close to 24 hours then we may be tempted to say that the clock mechanism somehow represents the length of the day. Millikan refers to such adaptations as tacit suppositions because they presuppose certain facts about the environment in order that their evolved function is fulfilled.

For a system to qualify as minimally representational, it must involve more than tacit suppositions. Firstly, there must be something identifiable as the representation itself: an "icon". Furthermore, the icon must have a "producer"

and a "consumer". It must be the function of the producer to generate the icon in accordance with a mapping rule that relates one or more dimensions of possible variance in the icon to variance in the environment. It must be the evolved function of the consumer to use or be guided by the icon in some way. If all of these conditions are met, Millikan suggests that the system involves an "intentional icon". For example, the waggle dance of the honeybee is a paradigm case of an intentional icon: the dance itself is the icon, the dancing bee is the producer, and a mapping rule relates the angle and duration of the dance to the direction and distance to a food source. The watching bees are the consumers of the icon, because it is the adaptive function of the dance to guide them to the food source. The important point is that there is a difference between tacitly supposing that the world — including your interaction partners — regularly works in a certain way, and evolving a distinct behaviour or trait that has been selected for on both sides (production and reception) precisely because it conveys information from one agent to the other.

Consider the difference between two scenarios. In the first



one, you are at home, and it is my job to pick you up in a car. I drive to your house: you are not outside, so I drive around the block repeatedly and check again for your presence each time I go by. Assuming I do that reasonably reliably, you can tacitly suppose the existence of my strategy, and go out into the street whenever you see my car going by. I will then see you and stop to pick you up on the next cycle. Both of us have strategies that rely on the other one acting a certain way but neither strategy has been exaggerated into a signal. We are coordinated but not communicative. The second scenario is exactly the same, except that through some adaptive process we have arrived at a communicative solution: I honk the horn three times in quick succession, and you come outside in response.

These two scenarios demonstrate the difficulty of showing that Quinn's (or our) observed behaviours are anything more than coordinated. Each agent is tacitly supposing that the other will rotate, align, move forwards and/or backwards, etc. The dance-like movement is, on the surface, reminiscent of the bee dance, and we suspect this resemblance has made many readers of Quinn's original paper confidently interpret the behaviour as communication. However, it is important to note that there is no mapping rule and no clear referential signalling going on.

What would it take to make Quinn's negotiation dance a signal? In answering this question, Millikan would agree closely with the ethologists. Non-signalling behaviours must provide the seed for signalling behaviours — how could it be otherwise? So the thing to look for in classifying something as "real communication" is a history of selection for exaggeration on both sides, both in the production of the signal and the sensitivity or scale of the response. In Quinn's paradigm we do not really see this: as far as we can tell from historical analyses of our runs, the agents hit on their coordination strategy and it remains essentially unchanged.

Quinn's dance in its current form appears to be a borderline case: it surely qualifies as an intention movement, and is quite possibly ripe for exaggeration into a signal. In the next section we try to push things towards communication by adding both perceptual noise and internal state to the agents. Noise may make a difference in that we can imagine the "dance signal" being exaggerated or strengthened to make sure it cannot be misunderstood in a noisy environment. State is a slightly different story: our stateless agents are necessarily reactive. It is not clear whether Quinn's CTRNN agents had any internal state; they might have, due to the possibility of recurrent connections. If we add state bits and find that this improves performance, that means that the task was "state-hungry", which in turn suggests a potential interpretation in terms of intentional icons, i.e., that the agents could be communicating about their current internal state.

## Results of the extended model

We extended our replication of Quinn's model to try to assess whether or not the evolved behaviours really qualify as communicative. In order to do so we have added two new features: perceptual noise and internal state. The addition of Gaussian noise to the sensory inputs adds ambiguity to the perceptual world of the agents and would seem likely to make the task more difficult. Thus it might be a driver for more explicitly communicative strategies. The second extension is the addition of 1 and then 2 bits of internal state to the agents. The acquisition of internal state enhances the cognitive capabilities of the agents, giving them more behavioural options than a purely reactive agent. This should make it easier for the agents to sequence their coordinated behaviours over time, but for our purposes it may also give them something to communicate *about*, i.e., their current internal state values.

We added 17 new sets of 30 runs each, employing different noise values (0%, 1%, 2%, 4%, 8% and 16%) and adding either 0, 1 or 2 bits of internal state to the agents. In the end we have a total of 18 run sets (including the original noiseless-stateless run) exploring every combination of noise and number of state bits. In order to reflect the increased range of behavioural possibilities that come with having internal state, we have also increased the number of classifiers from 5 (in stateless runs) to 10 (for 1-bit state runs) and 15 (for 2-bit state runs).

The results are presented in figure 3; the general pattern is in line with our expectations. We can see that the addition of noise decreases the performance of the agents in solving the task. On the other hand, the addition of state seems to make the task easier: the 2-bit condition is only slightly superior to the 1-bit condition, but both increases the mean fitness of the population over the stateless case.

When looking at the different runs individually, we find that state-equipped agents evolved more robust strategies than stateless ones. In fact, some of the 2-bit state solutions reach consistently optimal performance across the lower noise levels. In such cases, the mean fitness of the population reaches a sustained score of 1.0 with only occasional perturbations due to the randomness added by mutation. Since we have not observed such robust performance in any of the of the stateless runs we take this as evidence that the task chosen by Quinn, despite its simplicity, is "state-hungry".

In order to qualitatively assess whether or not the agents were evolving genuinely communicative solutions, we looked for the equivalent of Tinbergen's "intention movements" in the early stages of each evolutionary time line, and looked also for their ritualization or exaggeration into proper signals. We have found *some* suggestive cases of exaggeration in state-equipped runs, in particular for the forward-backwards movement that Quinn originally highlighted as a suspected signal. The movement sometimes becomes ex-

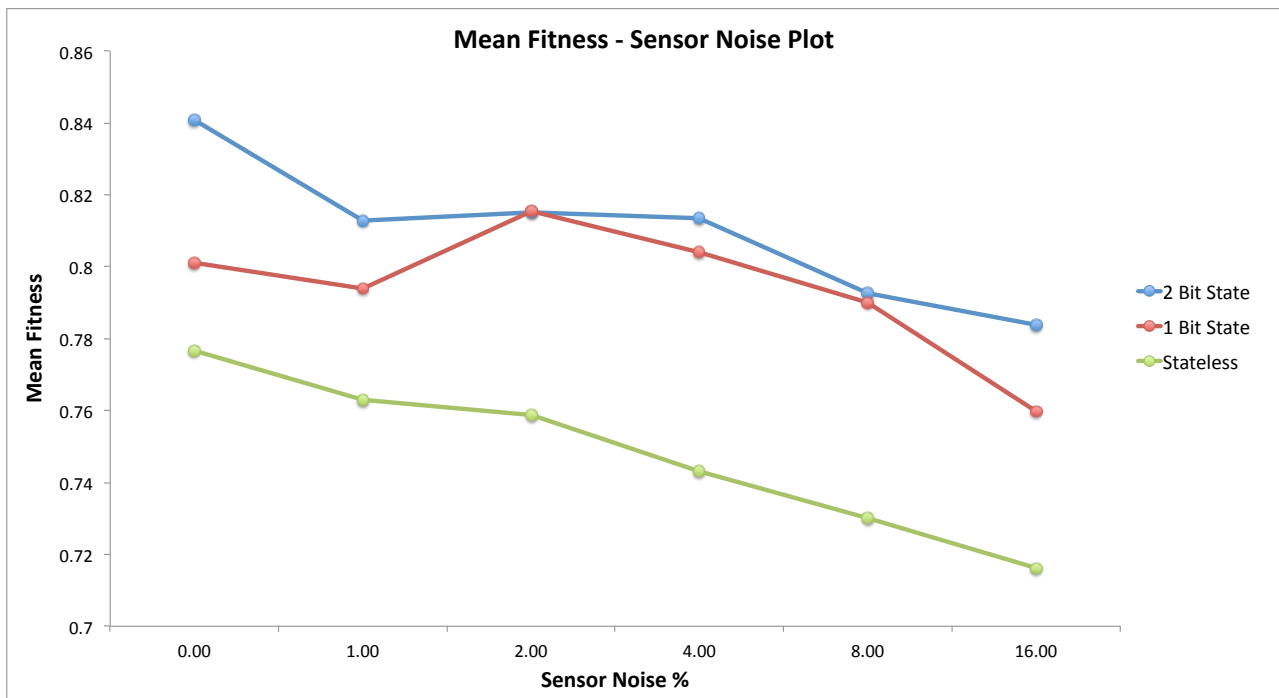


Figure 3: Results from the extended model. Mean fitness reached at the end of the evolutionary run is shown for various combinations of perceptual noise level and the number of state bits. Each plotted data point is an average across 30 replications. Standard errors across these replications are not shown for reasons of clarity, but the mean size of the standard error was 0.013. The general result is that performance is reduced as noise levels increase, and that at least one bit of state leads to better performance on the task.

aggregated just before the population starts to score perfect fitness scores. The exaggeration consists of a two-steps-forward-two-steps-backwards routine instead of the former one-step-forward-one-step-backwards. Despite this interesting result, we found no indications of a general trend. Furthermore, the runs that evolved such exaggerated “signals” were not among the runs with the highest overall average fitness (although, on the other hand, this kind of exaggeration never appeared in stateless runs). It may be that the task picked by Quinn is not “communication hungry”, i.e., it does not require explicit information transmission between the agents in order to achieve optimal performance levels (see Scheutz and Schermerhorn, 2008).

## Conclusions

We have achieved one of our goals, in demonstrating the generality of Quinn’s (2001) finding that sensory-motor interaction with no pre-defined communication channels can lead to coordinated behaviour. The result does not seem to be dependent on specific details of Quinn’s setup such as the CTRNN control architecture.

We also asked some critical questions as to whether the dance-like coordination behaviour should be seen as communicative. We extended Quinn’s model to include in-

creased levels of perceptual noise, and internal state for the agents. This was done with the intention of pushing the agents into developing exaggerated signalling and response behaviours over evolutionary time that would more clearly fit the definition of communication. Unsurprisingly, we found that higher levels of noise make the task more difficult. We also found that adding one or two bits of internal state improved performance, indicating that Quinn’s task is somewhat “state hungry”. Unfortunately we were not able to get consistent evidence of signal exaggeration and ritualization. We have to conclude that the dance-like coordination behaviour exhibited by the agents is at best a borderline case of true signalling.

The difficulty is that Quinn’s chosen task simply appears not to provide selective pressure for communication in Milikan’s sense of producing intentional icons. Scheutz and Schermerhorn (2008) have noted that this is true of many of the simple scenarios employed by ALife researchers. If we look at the world inhabited by our agents, it becomes clear that there is effectively not much to talk about: they always begin their interaction within sensor range of each other, the other agent is the only feature in the world and thus the only thing that can be detected by their sensors, and the cooperative goal of joint movement is always consistent. Once a co-

ordinated solution has been evolved, the agents are already performing near-optimally and there is no evolutionary pressure towards any exaggeration of the signal. We suspect that a promising direction for future work in this area is to use tasks in which referential communication about distant objects is essential for optimal performance (see e.g., Williams et al., 2008).

Why does all this matter anyway? Coordination or communication — what’s the difference? We believe it matters because there are two very different messages one can take from Quinn’s original finding. On the one hand you may see Quinn’s result as showing how the *appearance* of communication can be explained away as being just the result of mechanical feedback loops in a physical system. Some “enactivist” thinkers in ALife appear to endorse this position. The hope is to eventually demonstrate that human-level intelligence is really made up of a toolkit of sensory-motor tricks and hacks; Beer’s (2003) dynamical systems approach is a good example.

On the other hand, Quinn’s result can be seen as an attempt to bridge two levels of description. Quinn published his paper out of frustration with previous ALife work on signalling that constantly presupposed the very thing it was trying to explain, but that does not mean that he hoped to render talk of signals and channels irrelevant. If a model like Quinn’s could successfully show that communication can indeed emerge from sensory-motor interactions, we could take that not as undermining the concept of communication but as explaining how one level of description (L2: that of signals, symbols, and representations) can emerge from another (L1: the mechanics of sensory-motor feedback).

It has been argued (de Pinedo and Noble, 2008) that in explaining the behaviour of evolved agents, both agent- and sub-agent-level explanations will be necessary — and models like Quinn’s seem a useful step in that direction. Having established that L1 can give rise to L2, we thus establish that every subsequent simulation which incorporates L2-type communication does not need to provide direct evidence of the origins of that communication — we are safe in assuming that said communication would evolve in some fashion or other. Models like Quinn’s can thus provide *bridging explanations*: they verify the relationship between L1 and L2 and then allow those interested in L2 alone to get on with simulating phenomena at that level, confident that L2’s origins are understood.

The question as to which of these two views of communication and reference will ultimately prevail is of course still open. However, we are convinced that ALife simulations such as Quinn’s provide a uniquely valuable testing ground for working out the consequences of either approach.

## Acknowledgements

JA was supported by the UK’s Engineering and Physical Sciences Research Council via Southampton’s Institute for Complex Sys-

tems Simulation and doctoral training centre. JN was partly funded by research project FFI2010-19455 from the Spanish Ministry of Science. ES was partly supported by the EPSRC-funded *Care Life Cycle* project.

## References

- Beer, R. D. (2003). The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, 11(4):209–244.
- Bullock, S., Noble, J., Watson, R., and Bedau, M. A., editors (2008). *ALife XI: Proceedings of the Eleventh International Conference on Artificial Life*. MIT Press.
- de Pinedo, M. and Noble, J. (2008). Beyond persons: Extending the personal / subpersonal distinction to non-rational animals and artificial agents. *Biology and Philosophy*, 23(1):87–100.
- Lakoff, G. and Johnson, M. (1980). *Metaphors We Live By*. University of Chicago Press.
- Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors (1992). *Artificial Life II*. Addison-Wesley, Redwood City, CA.
- MacLennan, B. (1992). Synthetic ethology: An approach to the study of communication. In Langton et al. (1992), pages 631–658.
- Maynard Smith, J. and Szathmáry, E. (1995). *The Major Transitions in Evolution*. W. H. Freeman, Oxford.
- Millikan, R. G. (1984). *Language, Thought, and Other Biological Categories*. MIT Press / Bradford Books, Cambridge, MA.
- Noble, J. and Cliff, D. (1996). On simulating the evolution of communication. In Maes, P., Mataric, M., Meyer, J.-A., Pollack, J., and Wilson, S. W., editors, *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pages 608–617. MIT Press, Cambridge, MA.
- Quinn, M. (2001). Evolving communication without dedicated communication channels. In Kelemen, J. and Sosik, P., editors, *Advances in Artificial Life: Sixth European Conference on Artificial Life (ECAL’01)*, Lecture Notes in Artificial Intelligence, pages 357–366. Springer, Berlin.
- Scheutz, M. and Schermerhorn, P. (2008). The limited utility of communication in simple organisms. In Bullock et al. (2008), pages 521–528.
- Shannon, C. E. and Weaver, W. (1949). *The Mathematical Theory of Communication*. University of Illinois Press, Urbana.
- Tinbergen, N. (1964). The evolution of signalling devices. In Etkin, W., editor, *Social Behavior and Organization Among Vertebrates*, pages 206–230. University of Chicago Press, Chicago.
- Werner, G. M. and Dyer, M. G. (1992). Evolution of communication in artificial organisms. In Langton et al. (1992), pages 659–687.
- Williams, P. L., Beer, R. D., and Gasser, M. (2008). Evolving referential communication in embodied dynamical agents. In Bullock et al. (2008), pages 702–709.

# Detecting emergent processes in cellular automata with excess information

David Balduzzi<sup>1</sup>

<sup>1</sup> Department of Empirical Inference, MPI for Intelligent Systems, Tübingen, Germany  
david.balduzzi@tuebingen.mpg.de

## Abstract

Many natural processes occur over characteristic spatial and temporal scales. This paper presents tools for (i) flexibly and scalably coarse-graining cellular automata and (ii) identifying which coarse-grainings express an automaton's dynamics well, and which express its dynamics badly. We apply the tools to investigate a range of examples in Conway's Game of Life and Hopfield networks and demonstrate that they capture some basic intuitions about emergent processes. Finally, we formalize the notion that a process is emergent if it is better expressed at a coarser granularity.

## Introduction

Biological systems are studied across a range of spatiotemporal scales – for example as collections of atoms, molecules, cells, and organisms (Anderson, 1972). However, not all scales express a system's dynamics equally well. This paper proposes a principled method for identifying which spatiotemporal scale best expresses a cellular automaton's dynamics. We focus on Conway's Game of Life and Hopfield networks as test cases where collective behavior arises from simple local rules.

Conway's Game of Life is a well-studied artificial system with interesting behavior at multiple scales (Berlekamp et al., 1982). It is a 2-dimensional grid whose cells are updated according to deterministic rules. Remarkably, a sufficiently large grid can implement any deterministic computation. Designing patterns that perform sophisticated computations requires working with distributed structures such as gliders and glider guns rather than individual cells (Dennett, 1991). This suggests grid computations may be better expressed at coarser spatiotemporal scales.

The first contribution of this paper is a coarse-graining procedure for expressing a cellular automaton's dynamics at different scales. We begin by considering cellular automata as collections of spacetime coordinates termed occasions (cell  $n_i$  at time  $t$ ). Coarse-graining groups occasions into structures called *units*. For example a unit could be a  $3 \times 3$  patch of grid containing a glider at time  $t$ . Units do not have to be adjacent to one another; they interact through

*channel* – transparent occasions whose outputs are marginalized over. Finally, some occasions are set as *ground*, which fixes the initial condition of the coarse-grained system.

Gliders propagate at  $1/4$  diagonal squares per tic – the grid's "speed of light". Units more than  $4n$  cells apart cannot interact within  $n$  tics, imposing constraints on which coarse-grainings can express glider dynamics. It is also intuitively clear that units should group occasions concentrated in space and time rather than scattered occasions that have nothing to do with each other. In fact, it turns out that most coarse-grainings express a cellular automaton's dynamics badly.

The second contribution of this paper is a method for distinguishing good coarse-grainings from bad based on the following principle:

- *Coarse-grainings that generate more information, relative to their sub-grainings, better express an automaton's dynamics than those generating less.*

We introduce two measures to quantify the information generated by coarse-grained systems. Effective information,  $ei$ , quantifies how selectively a system's output depends on its input. Effective information is high if few inputs cause the output, and low if many do. Excess information,  $\xi$ , measures the difference between the information generated by a system and its subsystems.

With these tools in hand we investigate coarse-grainings of Game of Life grids and Hopfield networks and show that grainings with high  $ei$  and  $\xi$  capture our basic intuitions regarding emergent processes. For example, excess information distinguishes boring (redundant) from interesting (synergistic) information-processing, exemplified by blank patches of grid and gliders respectively.

Finally, the penultimate section converts our experience with examples in the Game of Life and Hopfield networks into a provisional formalization of the principle above. Roughly, we define a process as *emergent* if it is better expressed at a coarser scale.

The principle states that emergent processes are more than the sum of their parts – in agreement with many other approaches to quantifying emergence (Crutchfield, 1994;

Tononi, 2004; Polani, 2006; Shalizi and Moore, 2006; Seth, 2010). Two points distinguishing our approach from prior work are worth emphasizing. First, coarse-graining is *scalable*: coarse-graining a cellular automaton yields another cellular automaton. Prior works identify macro-variables such as temperature (Shalizi and Moore, 2006) or centre-of-mass (Seth, 2010) but do not show how to describe a system's dynamics purely in terms of these macro-variables. By contrast, an emergent coarse-graining is itself a cellular automaton, whose dynamics are computed via the mechanisms of its units and their connectivity (see below).

Second, our starting point is selectivity rather than predictability. Assessing predictability necessitates building a model and deciding what to predict. Although emergent variables may be robust against model changes (Seth, 2010), it is unsatisfying for emergence to depend on properties of *both* the process *and* the model. By contrast, effective and excess information depend only on the process: the mechanisms, their connectivity, and their output. A process is then emergent if its internal dependencies are best expressed at coarse granularities.

### Probabilistic cellular automata

**Concrete examples.** This paper considers two main examples of cellular automata: Conway's Game of Life and Hopfield networks (Hopfield, 1982).

The Game of Life is a grid of deterministic binary cells. A cell outputs 1 at time  $t$  iff: (i) three of its neighbors outputted 1s at  $t - 1$  or (ii) it and two neighbors outputted 1s at  $t - 1$ .

In a Hopfield network (Amit, 1989), cell  $n_k$  fires with probability proportional to

$$p(n_{k,t} = 1 | n_{\bullet,t-1}) \propto \exp \left[ \frac{1}{T} \sum_{j \rightarrow k} \alpha_{jk} \cdot n_{j,t-1} \right] \quad (1)$$

Temperature  $T$  controls network stochasticity. Attractors  $\{\xi^1, \dots, \xi^N\}$  are embedded into a network by setting the connectivity matrix as  $\alpha_{jk} = \sum_{\mu=1}^N (2\xi_j^\mu - 1)(2\xi_k^\mu - 1)$ .

**Abstract definition.** A *cellular automaton* is a finite directed graph  $X$  with vertices  $V_X = \{v_1 \dots v_n\}$ . Vertices are referred to as occasions; they correspond to *spacetime* coordinates in concrete examples. Each occasion  $v_l \in V_X$  is equipped with finite output alphabet  $A_l$  and Markov matrix (or *mechanism*)  $p_l(a_l | s_l)$ , where  $s_l \in S_l = \prod_{k \rightarrow l} A_k$ , the combined alphabet of the occasions targeting  $v_l$ . The mechanism specifies the probability that occasion  $v_l$  chooses output  $a_l$  given input  $s_l$ . The input alphabet of the entire automaton  $X$  is the product of the alphabets of its occasions  $X_{in} := \prod_{l \in V_X} A_l$ . The output alphabet is  $X_{out} = X_{in}$ .

*Remark.* The input  $X_{in}$  and output  $X_{out}$  alphabets are distinct copies of the same set. Inputs are causal interventions imposed via Pearl's  $do(-)$  calculus (Pearl, 2000). The probability of output  $a_l$  is computed via the Markov matrix:

$p_l(a_l | do(s_l))$ . The  $do(-)$  is not included in the notation explicitly to save space. However, it is always implicit when applying any Markov matrix.

A Hopfield network over time interval  $[\alpha, \beta]$  is an abstract automaton. Occasions are spacetime coordinates – e.g.  $v_l = n_{i,t}$ , cell  $i$  at time  $t$ . An edge connects  $v_k \rightarrow v_l$  if there is a connection from  $v_k$ 's cell to  $v_l$ 's and the time coordinates are  $t - 1$  and  $t$  respectively for some  $t$ . The mechanism is given by Eq. (1). Occasions at  $t = \alpha$ , with no incoming edges, can be set as fixed initial conditions or noise sources. Similar considerations apply to the Game of Life.

Non-Markovian automata (whose outputs depend on inputs over multiple time steps) have edges connecting occasions separated by more than one time step.

### Coarse-graining

Define a *subsystem*  $X$  of cellular automaton  $Y$  as a subgraph containing a subset of  $Y$ 's vertices and a subset of the edges targeting those vertices. We show how to coarse-grain  $X$ .

**Definition** (coarse-graining). *Let  $X$  be a subsystem of  $Y$ . The coarse-graining algorithm detailed below takes  $X \subset Y$  and data  $\mathcal{K}$  as arguments, and produces new cellular automaton  $X_{\mathcal{K}}$ . Data  $\mathcal{K}$  consists of (i) a partition of  $X$ 's occasions  $V_X = \mathbf{G} \cup \mathbf{C} \cup \mathbf{U}_1 \cup \dots \cup \mathbf{U}_N$  into ground  $\mathbf{G}$ , channel  $\mathbf{C}$  and units  $\mathbf{U}_1 \dots \mathbf{U}_N$  and (ii) ground output  $s^{\mathbf{G}}$ .*

Vertices of automaton  $X_{\mathcal{K}}$ , the new coarse-grained occasions, are units:  $V_{X_{\mathcal{K}}} := \{\mathbf{U}_1 \dots \mathbf{U}_N\}$ . The directed graph of  $X_{\mathcal{K}}$  is computed in Step 4 and the alphabets  $\mathbf{A}_l$  of units  $\mathbf{U}_l$  are computed in Step 5. Computing the Markov matrices (mechanisms) of the units takes all five steps.

The ground specifies occasions whose outputs are fixed: the initial condition  $s^{\mathbf{G}}$ . The channel specifies unobserved occasions: interactions between units propagate across the channel. Units are macroscopic occasions whose interactions are expressed by the coarse-grained automaton. Fig. 1 illustrates coarse-graining a simple automaton.

There are no restrictions on partitions. For example, although the ground is intended to provide the system's initial condition, it can contain any spacetime coordinates so that in pathological cases it may obstruct interactions between units. Distinguishing good coarse-grainings from bad is postponed to later sections.

**Algorithm.** Apply the following steps to coarse-grain:

**Step 1.** *Marginalize over extrinsic inputs.*

External inputs are treated as independent noise sources; we are only interested in *internal* information-processing. An occasion's input alphabet decomposes into a product  $S_l = S_l^X \times S_l^{Y \setminus X}$  of inputs from within and without the system. For each occasion  $v_l \in V_X$ , marginalize over external outputs using the uniform distribution:

$$p_l(a_l | s_l^X) := \sum_{S_l^{Y \setminus X}} p_l(a_l | s_l^X, s_l^{Y \setminus X}) \cdot p_{unif}(s_l^{Y \setminus X}). \quad (2)$$

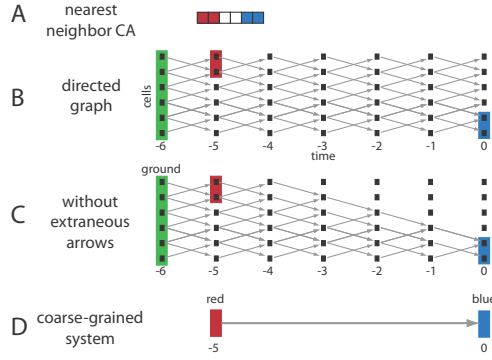


Figure 1: (A) An automaton of 6 cells connected to their immediate neighbors. (B): The directed graph of occasions over time interval  $[-6, 0]$ . Green occasions are ground. Red and blue occasions form two units. Other occasions are channel. (C): Edges whose signals do not reach the blue unit have no effect. (D): The coarse-grained system consists of two units (macro-occasions).

### Step 2. Fix the ground.

Ground outputs are fixed in the coarse-grained system. Graining  $\mathcal{K}$  imposes a second decomposition onto  $v_l$ 's input alphabet,  $S_l^X = S_l^G \times S_l^C \times S_l^U$  where  $U = \cup_k U_k$ . Subsume the ground into  $v_l$ 's mechanism by specifying

$$p_l^G(a_l | s_l^C, s_l^U) := p_l(a_l | s_l^G, s_l^C, s_l^U).$$

### Step 3. Marginalize over the channel.

The channel specifies transparent occasions. Perturbations introduced into units propagate through the channel until they reach other units where they are observed. Transparency is imposed by marginalizing over the channel occasions in the product mechanism

$$p_{\mathcal{K}}(x_{out}^{\mathcal{K}} | x_{in}^{\mathcal{K}}) := \sum_{l \in \mathbf{C}} \prod_{l \in \mathbf{C} \cup \mathbf{U}} p_l^G(x_{out}^l | x_{in}^l), \quad (3)$$

where superscripts denote that inputs and outputs are restricted, for  $\mathcal{K}$ , to occasions in units in  $\mathcal{K}$  (since channel is summed over and ground is already fixed) and, for each  $l$ , to the inputs and outputs of occasion  $v_l$ .

For example, consider cellular automaton with graph  $v_a \rightarrow v_b \rightarrow v_c$  and product mechanism  $p(c|b)p(b|a)p(a)$ . Setting  $v_b$  as channel and marginalizing yields coarse-grained mechanism  $\sum_b p(c|b)p(b|a)p(a) = p(c|a)p(a)$ . The channel is rendered transparent and new mechanism  $p(c|a)$  convolves  $p(c|b)$  and  $p(b|a)$ .

### Step 4. Compute the effective graph of coarse-graining $X_{\mathcal{K}}$ .

The micro-alphabet of unit  $U_l$  is  $\tilde{\mathbf{A}}_l := \prod_{k \in U_l} A_k$ . The mechanism of  $U_l$  is computed as in Eq. (3) with the product restricted to occasions  $j \in \mathbf{C} \cup U_l$ , thus obtaining  $p_{U_l}(a_l | x_{in})$  where  $a_l \in \tilde{\mathbf{A}}_l$ .

Two units  $U_k$  and  $U_l$  are connected by an edge if the outputs of  $U_k$  make a difference to the behavior of  $U_l$ . More

precisely, we draw an edge if  $\exists a_k, a'_k \in \tilde{\mathbf{A}}_k$  such that

$$p_{U_l}(a_l | \overline{x_{in}}, a_k) \neq p_{U_l}(a_l | \overline{x_{in}}, a'_k) \text{ for some } a_l \in \tilde{\mathbf{A}}_l.$$

Here,  $\overline{x_{in}}$  denotes the input from all units other than  $U_k$ .

The effective graph need not be acyclic. Intervening via the  $do(-)$  calculus allows us to work with cycles.

### Step 5. Compute macro-alphabets of units in $X_{\mathcal{K}}$ .

Coarse-graining can eliminate low-level details. Outputs that are distinguishable at the base level may not be after coarse-graining. This can occur in two ways. Outputs  $b$  and  $b'$  have indistinguishable effects if  $p(a|b, c) = p(a|b', c)$  for all  $a$  and  $c$ . Alternatively, two outputs react indistinguishably if  $p(b|c) = p(b'|c)$  for all  $c$ .

More precisely, two outputs  $u_l$  and  $u'_l$  of unit  $U_l$  are equivalent, denoted  $u_l \sim_{\mathcal{K}} u'_l$ , iff

$$p_{\mathcal{K}}(x_{out} | \overline{x_{in}}, u_l) = p_{\mathcal{K}}(x_{out} | \overline{x_{in}}, u'_l) \text{ and} \\ p_{U_l}(u_l | x_{in}^{\mathcal{K}}) = p_{U_l}(u'_l | x_{in}^{\mathcal{K}}) \text{ for all } x_{out}, x_{in}.$$

Picking a single element from each equivalence class obtains the macro-alphabet  $\mathbf{A}_l$  of the unit  $U_l$ . The mechanism of  $U_l$  is  $p_{U_l}$ , Step 4, restricted to macro-alphabets.

## Information

This section extends prior work to quantify the information generated by a cellular automaton, both as a whole and relative to its subsystems (Balduzzi and Tononi, 2008, 2009).

Given subsystem  $m$  of  $X$ , let  $p_m(x_{out} | x_{in})$ , or  $m$  for short, denote its *mechanism* or Markov matrix. The mechanism is computed by taking the Markov matrix of each occasion in  $X$ , marginalizing over extrinsic inputs (edges not in  $X$ ) as in Eq. (2), and taking the product. It is notationally convenient to write  $p_m$  as though its inputs and outputs are  $x_{out}$  and  $x_{in}$ , even though  $m$  does not in general contain all occasions in  $X$  and therefore treats some inputs and outputs as extrinsic, unexplainable noise. We switch freely between terms “subsystem” and “submechanism” below.

**Effective information** quantifies how selectively a mechanism discriminates between inputs when assigning them to an output. Alternatively, it measures how sharp the functional dependencies leading to an output are.

The *actual repertoire*  $\hat{p}_m(X_{in} | x_{out})$  is the set of inputs that cause (lead to) mechanism  $m$  choosing output  $x_{out}$ , weighted by likelihood according to Bayes' rule

$$\hat{p}_m(x_{in} | x_{out}) := \frac{p_m(x_{out} | do(x_{in}))}{p(x_{out})} \cdot p_{unif}(x_{in}). \quad (4)$$

The  $do(-)$  notation and hat  $\hat{p}$  remind that we first *intervene* to impose  $x_{in}$  and then apply Markov matrix  $p_m$ .

For deterministic mechanisms, i.e. functions  $f : X_{in} \rightarrow X_{out}$ , the actual repertoire assigns  $\hat{p} = \frac{1}{|f^{-1}(x_{out})|}$  to elements of the pre-image and  $\hat{p} = 0$  to other elements of  $X_{in}$ .

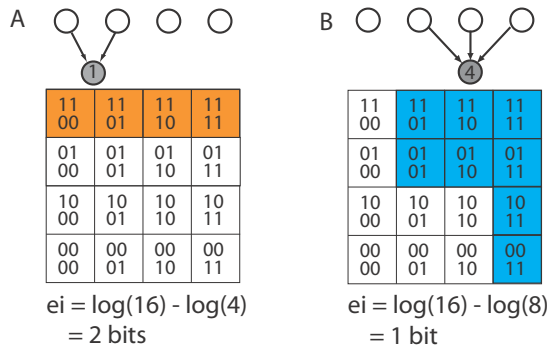


Figure 2: Categorization and information. Cells fire if they receive two or more spikes. The  $16 = 2^4$  possible outputs by the top layer are arranged in a grid. (AB): Cells  $n_1$  and  $n_4$  fire when the output is in the orange and blue regions respectively. Cell  $n_1$ 's response is more informative than  $n_4$ 's since it fires for fewer inputs.

The shaded regions in Fig. 2 show outputs of the top layer that cause the bottom cell to fire.

Effective information generated when  $m$  outputs  $x_{out}$  is Kullback-Leibler divergence ( $KL[p||q] = \sum_i p_i \log_2 \frac{p_i}{q_i}$ ),

$$ei(m, x_{out}) := KL[\hat{p}_m(X_{in}|x_{out}) || p_{unif}(X_{in})]. \quad (5)$$

Effective information is *not* a statistical measure: it depends on the mechanism and a *particular* output  $x_{out}$ .

Effective information generated by deterministic function  $f$  is  $ei(f, x_{out}) = \log_2 \frac{|X_{in}|}{|f^{-1}(x_{out})|}$  where  $|\cdot|$  denotes cardinality. In Fig. 2,  $ei$  is the logarithm of the ratio of the total number of squares to the number of shaded squares.

**Excess information** quantifies how much more information a mechanism generates than the sum of its submechanisms – how synergistic the internal dependencies are.

Given subsystem with mechanism  $m$ , partition  $\mathcal{P} = \{M^1 \dots M^m\}$  of the occasions in  $src(m)$ , and output  $x_{out}$ , define excess information as follows. Let  $m^j := m \cap (M^j \times X)$  be the restriction of  $m$  to sources in  $M^j$ . *Excess information over  $\mathcal{P}$  is*

$$\xi(m, \mathcal{P}, x_{out}) := ei(m, x_{out}) - \sum_j ei(m^j, x_{out}). \quad (6)$$

Excess information (sans partition) is computed over the information-theoretic weakest link  $\mathcal{P}^{MIP}$

$$\xi(m, x_{out}) := \xi(m, \mathcal{P}^{MIP}, x_{out}). \quad (7)$$

Let  $A_{M^j} := \prod_{l \in M^j} A_l$ . The minimum information partition<sup>1</sup>  $\mathcal{P}^{MIP}$  minimizes normalized excess information:

$$\mathcal{P}^{MIP} := \arg \min_{\mathcal{P}} \frac{\xi(m, \mathcal{P}, x_{out})}{\mathcal{N}_{\mathcal{P}}}, \text{ where}$$

$$\mathcal{N}_{\mathcal{P}} := (m-1) \cdot \min_j \{\log_2 |A_{M^j}|\}.$$

<sup>1</sup>We restrict to bipartitions to reduce the computational burden.

Excess information is negative if any decomposition of the system generates more information than the whole.

Fig. 3 shows how two cells taken together can generate the same, less, or more information than their sum taken individually depending on how their categorizations overlap. Note the figure decomposes the mechanism of the system over *targets* rather than sources and so does not depict excess information – which is more useful but harder to illustrate.

Effective information and excess information can be computed for any submechanism of any coarse-graining of any cellular automaton.

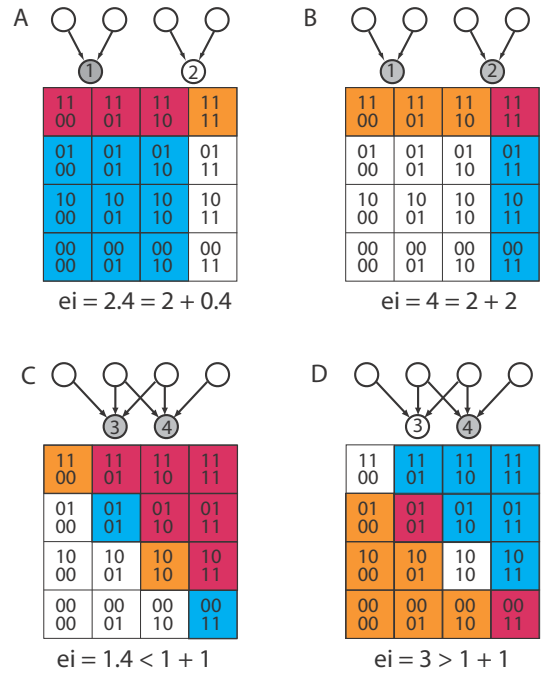


Figure 3: Independent, redundant and synergistic information. (AB): Independent. Orthogonal categorizations, orange+pink and blue+pink shadings respectively, by  $n_1$  and  $n_2$ . (C): Partially redundant. Both cells fire; categorizations overlap (pink) more “than expected” and  $ei(n_3 n_4, 11) < ei(n_3, 1) + ei(n_4, 1)$ . (D): Synergistic. Overlap is less “than expected”;  $ei(n_3 n_4, 01) > ei(n_3, 0) + ei(n_4, 1)$ .

## Application: Conway’s Game of Life

The Game of Life has interesting dynamics at a range of spatiotemporal scales. At the atomic level, each coordinate (cell  $i$  at time  $t$ ) is an occasion and information processing is extremely local. At coarser granularities, information can propagate through channels, so that units generate information at a distance. Gliders, for example, are distributed objects that can interact over large distances in space and time, Fig. 4A, and provide an important example of an emergent process (Dennett, 1991; Beer, 2004).

This section shows how effective and excess information quantifiably distinguish coarse-grainings expressing glider



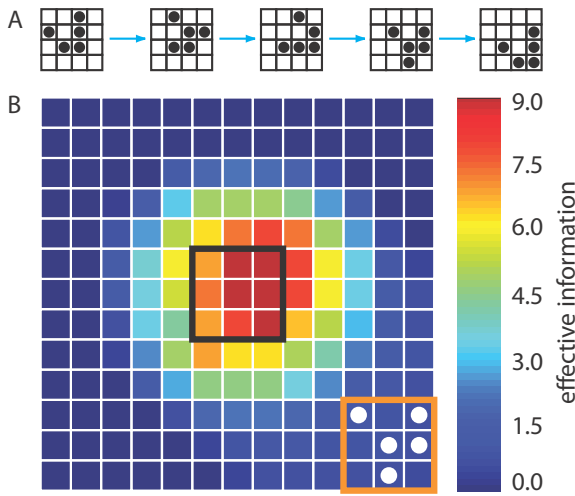


Figure 4: Detecting focal points. (A): A glider moves 1 diagonal square every 4 time steps. (B): Cells in the orange and black outlined  $3 \times 3$  squares are units at  $t = 0$  and  $t = -20$  respectively, with  $x_{out}$  the glider shown. Cells at  $t = -21$  are blank ground; other occasions are channel. Shifting the position of the black square produces a family of coarse-grainings. Effective information is shown as the black square's center varies over the grid.

dynamics well from those expressing it badly.

**Effective information detects focal points.** Fig. 4A shows a glider trajectory, which passes through 1 diagonal step over 4 tics. Fig. 4B investigates how glider trajectories are captured by coarse-grainings: if there is a glider in the  $3 \times 3$  orange square at time 0, Fig. 4B, it must have passed through the black square at  $t = -20$  to get there. Are coarse-grainings that respect glider trajectories quantifiably better than those that do not?

Fig. 4B fixes occasions in the black square at  $t = -20$  and the orange square at  $t = 0$  as units (18 total), the ground as blank grid at  $t = -21$  and everything else as channel. Varying the spatial location of the black square over the grid, we obtain a *family* of coarse-grainings. Effective information for each graining in the family is shown in the figure. There is a clear focal point exactly where the black square intersects the spatiotemporal trajectory of the glider where *ei* is maximized (dark red). Effective information is zero for locations that are too far or too close at  $t = -20$  to effect the output of the orange square at  $t = 0$ .

Effective information thus provides a tool analogous to a camera focus: grainings closer to the focal point express glider dynamics better.

**Macroscopic texture varies with distance.** The behavior of individual cells within a glider trajectory is far more complicated than the glider itself, which transitions through 4 phases as it traverses its diagonal trajectory, Fig. 4A. Does coarse-graining quantifiably simplify dynamics?

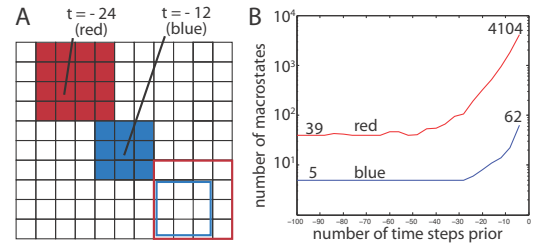


Figure 5: Macro-alphabets as a function of distance. (A): Consider two families of coarse-grainings with channel and ground as in Fig. 4. First, take the blue squares (filled and empty) as units at times  $-4n$  and 0 where  $n$  is the diagonal distance between them. Second, repeat for the red squares. (B): Log-plot of the size of the filled squares' macro-alphabets as a function of  $-4n$ .

Fig. 5 constructs pairs of  $3 \times 3$  units out of occasions at various distances from one another and computes their macro-alphabets. A  $3 \times 3$  unit has a micro-alphabet of  $2^9 = 512$  outputs. The macro-alphabet is found by grouping micro-outputs together into equivalence classes if their effect is the same after propagating through the channel. We find that the size of the macro-alphabet decreases exponentially as the distance between units increases, stabilizing at 5 macro-outputs: the 4 glider phases in Fig. 4A and a large equivalence class of outputs that do not propagate to the target unit and are equivalent to a blank patch of grid. A similar phenomenon occurs for pairs of  $4 \times 4$  units, also Fig. 5.

Continuing the camera analogy: at close range the texture of units is visible. As the distance increases, the channel absorbs more of the detail. The computational texture of the system is simpler at coarser-grains yielding a more symbolic description where glider dynamics are described via 4 basic phases produced by a single macroscopic unit rather than  $2^9$  outputs produced by 9 microscopic occasions.

**Excess information detects spatial organization.** So far we have only considered grainings of the Game of Life that respect its spatial organization – in effect, taking the spatial structure for granted. *A priori*, there is nothing stopping us from grouping the 8 gray cells in Fig. 6A into a single unit that *does not* respect the spatial organization, since its constituents are separated in space. Are coarse-grainings that respect the grid-structure quantifiably better than others?

Fig. 6A shows a coarse-graining that does *not* respect the grid. It constructs two units, one from *both* gray squares at  $t = 1$  and the other from *both* red squares at  $t = 0$ . Intuitively, the coarse-graining is unsatisfactory since it builds units whose constituent occasions have nothing to do with each other over the time-scale in question. Quantitatively, excess information over the obvious partition  $\mathcal{P}$  of the system into two parts is 0 bits. It is easy to show  $\xi \leq 0$  for any disjoint units. By comparison, the coarse-grainings in panels CD, which respect the grid structure, both generate positive excess information.



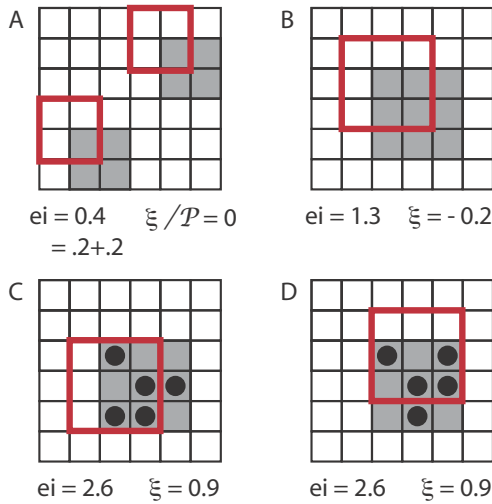


Figure 6: Detecting spatial organization. Units are the cells in the red (thick-edged) and gray (filled) squares at  $t = 0$  and  $t = 1$  respectively; other occasions are extrinsic noise. (A):  $\xi = 0$ . The coarse-graining groups non-interacting occasions into units. (B):  $\xi < 0$ . A blank grid is highly redundant. (CD):  $\xi > 0$ . Gliders perform interesting information-processing.

Thus we find that not only does our information-theoretic camera have an automatic focus, it also detects when processes hang together to form a single coherent scene.

**Excess information detects gliders.** Blank stretches of grid, Fig. 6B, are boring. There is nothing going on. Are interesting patches of grid quantifiably distinguishable from boring patches?

Excess information distinguishes blank grids from gliders:  $\xi$  on the blank grid is negative, Fig. 6B, since the information generated by the cells is redundant analogous to Fig. 3C. By contrast,  $\xi$  for a glider is positive, Fig. 6CD, since its cells perform synergistic categorizations, similarly to Fig. 3D. Glider trajectories are also captured by excess information: varying the location of the red units (at  $t = 0$ ) around the gray units we find that  $\xi$  is maximized in the positions shown, Fig. 6CD, thus expressing the rightwards and downwards motions of the respective gliders.

Returning to the camera analogy, blank patches of grid fade into (back)ground or are (transparent) channel, whereas gliders are highlighted front and center as units.

### Application: Hopfield networks

Hopfield networks embed energy landscapes into their connectivity. For any initial condition they tend to one of few attractors – troughs in the landscape (Hopfield, 1982; Amit, 1989). Although cells in Hopfield networks are quite different from neurons, there is evidence suggesting neuronal populations transition between coherent distributed states similar to attractors (Abeles et al., 1995; Jones et al., 2007).

$t$	output		INT: $B \rightarrow B$		EXT: $A \rightarrow B$	
	$A$	$B$	$ei$	$\max \xi$	$ei$	$\max \xi$
0	00000000	01010101				
1	10100011	01010101	<b>2.42</b>	<b>0.10</b>	0.31	0.04
2	10101010	00010101	1.85	0.08	<b>2.44</b>	<b>0.16</b>
3	10101010	00101011	1.96	0.12	<b>6.89</b>	<b>0.27</b>
4	10101010	00101010	<b>1.85</b>	0.08	1.60	<b>0.10</b>
5	10101010	10101010	<b>2.42</b>	<b>0.10</b>	0.90	0.06
6	10101010	10101010	<b>2.42</b>	<b>0.10</b>	0.31	0.04

Table 1: Analysis of unidirectionally coupled Hopfield networks  $A \rightarrow B$  each containing 8 cells. The networks and coupling embed attractors  $\{00001111, 00110011, 01010101\}$  and their mirrors. Temperature is  $T = 0.25$ . A sample run is analyzed using two coarse-grainings: INT captures  $B$ 's effect on itself and EXT captures  $A$ 's effect on  $B$ ; see text.

Attractors are population level phenomena. They arise because of interactions between groups of cells – no single cell is responsible for their existence – suggesting that coarse-graining may reveal interesting features of attractor dynamics.

**Effective information detects causal interactions.** Table 1 analyzes a sample run of unidirectionally coupled Hopfield networks  $A \rightarrow B$ . Network  $A$  is initialized at an unstable point in the energy landscape and  $B$  in an attractor.  $A$  settles into a different attractor from  $B$  and then shoves  $B$  into the new attractor over a few time steps. Intuitively,  $A$  only exerts a strong force on  $B$  once it has settled in an attractor and before  $B$  transitions to the same attractor. Is the force  $A$  exerts on  $B$  quantitatively detectable?

Table 1 shows the effects of  $A$  and  $B$  respectively on  $B$  by computing  $ei$  for two coarse-grainings constructed for each transition  $t \rightarrow t + 1$ . Coarse-graining INT sets cells in  $B$  at  $t$  and  $t + 1$  as units and  $A$  as extrinsic noise. EXT sets cells in  $A$  at  $t$  and  $B$  at  $t + 1$  as units and fixes  $B$  at time  $t$  as ground.

INT generates higher  $ei$  for all transitions except  $1 \rightarrow 2 \rightarrow 3$ , precisely when  $A$  shoves  $B$ . Effective information is high when an output is sensitive to changes in an input so it is unsurprising that  $B$  is more sensitive to changes in  $A$  exactly when  $A$  forces  $B$  out from one attractor into another. Analyzing other sample runs (not shown) confirms that  $ei$  reliably detects when  $A$  shoves  $B$  out of an attractor.

**Macroscopic mechanisms depend on the ground.** Fixing the ground incorporates population-level biases into a coarse-grained cellular automaton's information-processing.

The ground in coarse-graining EXT (i.e. the output of  $B$  at  $t - 1$ ) biases the mechanisms of the units in  $B$  at time  $t$ . When the ground is an attractor, it introduces tremendous inertia into the coarse-grained dynamics since  $B$  is heavily biased towards outputting the attractor again. Few inputs from  $A$  can overcome this inertia, so if  $B$  is pushed out of an attractor it generates high  $ei$  about  $A$ . Conversely, when  $B$  stays in an attractor, e.g. transition  $5 \rightarrow 6$ , it follows its internal bias and so generates low  $ei$  about  $A$ .

**Excess information detects attractor redundancy.** Following our analysis of gliders, we investigate how attractors are captured by excess information. It turns out that  $\xi$  is negative in all cases: the functional dependencies within Hopfield networks are redundant. An attractor is analogous to a blank Game of Life grid where little is going on. Thus, although attractors are population-level phenomena, we exclude them as emergent processes.

**Excess information expresses attractor transitions.** We therefore refine our analysis and compute the subset of units at time  $t$  that maximize  $\xi$ ; maximum values are shown in Table 1. We find that the system decomposes into pairs of occasions with low  $\xi$ , except when  $B$  is shoved, in which case larger structures of 5 occasions emerge. This fits prior analysis showing transitions between attractors yield more integrated dynamics (Balduzzi and Tononi, 2008) and suggestions that cortical dynamics is metastable, characterized by antagonism between local attractors (Friston, 1997).

Our analysis suggests that *transitions* between attractors are the most interesting emergent behaviors in coupled Hopfield networks. How this generalizes to more sophisticated models remains to be seen.

## Emergence

The examples show we can quantify how well a graining expresses a cellular automaton's dynamics. Effective information detects glider trajectories and also captures when one Hopfield network shoves another. However,  $ei$  does not detect whether a unit is integrated. For this we need excess information, which compares the information generated by a mechanism to that generated by its submechanisms. Forming units out of disjoint collections of occasions yields  $\xi = 0$ . Moreover, boring units (such as blank patches of grid or dead-end fixed point attractors) have negative  $\xi$ . Thus,  $\xi$  is a promising candidate for quantifying emergent processes.

This section formalizes the intuition that a system is emergent if its dynamics are better expressed at coarser spatiotemporal granularities. The idea is simple. Emergent units should generate more excess information, and have more excess information generated about them, than their sub-units. Moreover emergent units should generate more excess information than *neighboring* units, recall Fig. 4.

Stating the definition precisely requires some notation. Let  $\text{src}_{v_l} = \{v_l\} \cup \{v_k | k \rightarrow l\}$  and similarly for  $\text{trg}_{v_l}$ . Let  $\mathcal{J}$  be a subgraining of  $\mathcal{K}$ , denoted  $\mathcal{J} \prec \mathcal{K}$ , if for every  $\mathbf{U}_j \in \mathcal{J}$  there is a unit  $\mathbf{U}_k \in \mathcal{K}$  such that  $\mathbf{U}_j \subsetneq \mathbf{U}_k$ . We compare mechanism  $\mathbf{m} \subset \mathcal{K}$  with its subgrains via

$$\xi_{\mathcal{K}/\mathcal{J}}(\mathbf{m}, x_{out}) := ei_{\tilde{\mathcal{K}}}(\mathbf{m}, x_{out}) - \sum_{v_j \in \mathcal{J}} ei_{\tilde{\mathcal{J}}}(\mathbf{m}^j, x_{out}),$$

where  $\mathbf{m}^j = \mathbf{m} \cap \text{src}_{v_j}$  and  $ei_{\tilde{\mathcal{K}}}$  signifies effective information is computed over  $\mathcal{K}$  using micro-alphabets.

**Definition (emergence).** Fix cellular automaton  $X$  with output  $x_{out}$ . Coarse-graining<sup>2</sup>  $\mathcal{K}$  is emergent if it satisfies conditions E1 and E2.

E1. Each unit  $\mathbf{U}_l \in \mathcal{K}$  generates excess information about its sources and has excess information generated about it by its targets, relative to subgrains  $\mathcal{J} \prec \mathcal{K}$ :

$$0 < \xi_{\mathcal{J}/\mathcal{K}}(\text{src}_{\mathbf{U}_l}, x_{out}) \text{ and } 0 < \xi_{\mathcal{J}/\mathcal{K}}(\text{trg}_{\mathbf{U}_l}, x_{out}). \quad (8)$$

E2. There is an *emergent* subgrain  $\mathcal{J} \prec \mathcal{K}$  such that (i) every unit of  $\mathcal{K}$  contains a unit of  $\mathcal{J}$  and (ii) neighbors  $\mathcal{K}'$  (defined below) of  $\mathcal{K}$  with respect to  $\mathcal{J}$  satisfy

$$\xi_{\mathcal{J}/\mathcal{K}'}(\text{src}_{\mathbf{U}'}, x_{out}) \leq \xi_{\mathcal{J}/\mathcal{K}}(\text{src}_{\mathbf{U}}, x_{out}) \quad (9)$$

for all  $\mathbf{U} \in \mathcal{K}$ , and similarly for  $\text{trg}$ 's.

If  $\mathcal{K}$  has no emergent subgrains then E2 is vacuous.

Grain  $\mathcal{K}'$  is a *neighbor* of  $\mathcal{K}$  with respect to  $\mathcal{J} \prec \mathcal{K}$  if for every  $\mathbf{U} \in \mathcal{K}$  there is a unique  $\mathbf{U}' \in \mathcal{K}'$  satisfying

- N1. there is a unit  $T \in \mathcal{J}$  such that  $T \subset \mathbf{U}, \mathbf{U}', \text{src}_T \subset \text{src}_{\mathbf{U}}, \text{src}_{\mathbf{U}'}$  and similarly for  $\text{trg}$ ; and
- N2. the alphabet of  $\mathbf{U}'$  is no larger than  $\mathbf{U}$ :  $|\prod_{k \in \mathbf{U}'} A_k| \leq |\prod_{l \in \mathbf{U}} A_l|$ , and similarly for the combined alphabets of their sources and targets respectively.

The graining  $\mathcal{E}_X$  that best expresses  $X$  outputting  $x_{out}$  is found by maximizing normalized excess information:

$$\mathcal{E}_X(x_{out}) := \arg \max_{\{\mathcal{K} | \text{emergent}\}} \frac{\xi(\mathcal{K}, x_{out})}{\mathcal{N}_{\mathcal{P}_{MIP}}^{\mathcal{K}}}. \quad (10)$$

Here,  $\mathcal{N}_{\mathcal{P}_{MIP}}^{\mathcal{K}}$  is the normalizing constant found when computing the minimum information partition for  $\mathcal{K}$ .

**Some implications.** We apply the definition to the Game of Life to gain insight into its mechanics.

Condition E1 requires that interactions between units and their sources (and targets) are synergistic, Fig. 6CD. Units that decompose into independent pieces, Fig. 6A, or perform highly redundant operations, Fig. 6B, are therefore not emergent.

Condition E2 compares units to their neighbors. Rather than build the automaton's spatial organization directly into the definition, neighbors of  $\mathcal{K}$  are defined as coarse-grainings whose units overlap with  $\mathcal{K}$  and whose alphabets are no bigger. Coarse-grainings with higher  $\xi$  than their neighbors are closer to focal points, recall Fig. 4 and Fig. 6CD, where  $\xi$  was maximized for units respecting glider trajectories. An analysis of glider boundaries similar in spirit to this paper is (Beer, 2004).

<sup>2</sup>Ground output  $s^G$  is  $x_{out}$  restricted to ground occasions.

Finally, Eq. (10) picks out the most expressive coarse-graining. The normalization plays two roles. First, it biases the optimization towards grainings whose MIPs contain few, symmetric parts following (Balduzzi and Tononi, 2008). Second, it biases the optimization towards systems with simpler macro-alphabets. Recall, Fig. 5, that coarse-graining produces more symbolic interactions by decreasing the size of alphabets. Simplifying alphabets typically reduces effective and excess information since there are less bits to go around. The normalization term rewards simpler levels of description, so long as they use the bits in play more synergistically.

## Discussion

In this paper we introduced a flexible, scalable coarse-graining method that applies to any cellular automaton. Our notion of automaton applies to a broad range of systems. The constraints are that they (i) decompose into discrete components with (ii) finite alphabets where (iii) time passes in discrete tics. We then described how to quantify the information generated when a system produces an output (at any scale) both as a whole and relative to its subsystems. An important feature of our approach is that the output  $x_{out}$  of a graining is incorporated into the ground and also directly influences  $ei$  and  $\xi$  through computation of the actual repertoires. Coarse-graining and emergence therefore capture some of the *suppleness* of biological processes (Bedau, 1997): they are context-dependent and require many *ceteris paribus* clauses (i.e. background) to describe.

Investigating examples taken from Conway's Game of Life and coupled Hopfield networks, we accumulated a small but significant body of evidence confirming the principle that *expressive coarse-grainings generate more information relative to sub-grainings*. Finally, we provisionally defined emergent processes. The definition is provisional since it derives from analyzing a small fraction of the possible coarse-grainings of only two kinds of cellular automata.

Hopfield networks and the Game of Life are simple models capturing some important aspects of biological systems. Ultimately, we would like to analyze emergent phenomena in more realistic models, in particular of the brain. Conscious percepts take 100-200ms to arise and brain activity is (presumably) better expressed as comparatively leisurely interactions between neurons or neuronal assemblies rather than much faster interactions between atoms or molecules (Tononi, 2004). To apply the techniques developed here to more realistic models we must confront a computational hurdle: the number of coarse-grainings that can be imposed on large cellular automata is vast. Nevertheless, the approach developed here may still be of use. First, manipulating macro-alphabets provides a method for performing approximate computations on large-scale systems. Second, for more fine-grained analysis, initial estimates about which coarse-grainings best express a system's dynamics can be

fine-tuned by comparing them with neighbors.

**Acknowledgements.** The author thanks Dominik Janzing for many useful comments on an earlier draft, Giulio Tononi for stimulating conversations and Virgil Griffiths for emphasizing the importance of excess information.

## References

- Abeles, M., Bergman, H., Gat, I., Meilijson, I., Seidemann, E., Tishby, N., and Vaadia, E. (1995). Cortical activity flips among quasi-stationary states. *Proc. Nat. Acad. Sci.*, 92:8616–8620.
- Amit, D. (1989). *Modelling brain function: the world of attractor neural networks*. Cambridge University Press.
- Anderson, P. W. (1972). More is different. *Science*, 177(4047):393–6.
- Balduzzi, D. and Tononi, G. (2008). Integrated Information in Discrete Dynamical Systems: Motivation and Theoretical Framework. *PLoS Comput Biol*, 4(6):e1000091.
- Balduzzi, D. and Tononi, G. (2009). Qualia: the geometry of integrated information. *PLoS Comput Biol*, 5(8):e1000462.
- Bedau, M. A. (1997). Emergent models of supple dynamics in life and mind. *Brain Cogn*, 34(1):5–27.
- Beer, R. D. (2004). Autopoiesis and cognition in the game of life. *Artif Life*, 10(3):309–26.
- Berlekamp, E., Conway, J., and Guy, R. (1982). *Winning Ways for your Mathematical Plays*, volume 2. Academic Press.
- Crutchfield, J. (1994). The calculi of emergence: Computation, dynamics, and induction. *Physica D*, 75:11–54.
- Dennett, D. C. (1991). Real Patterns. *J. Philosophy*, 88(1):27–51.
- Friston, K. (1997). Transients, metastability and neuronal dynamics. *Neuroimage*, 5:164–171.
- Hopfield, J. (1982). Neural networks and physical systems with emergent computational properties. *Proc. Nat. Acad. Sci.*, 79:2554–2558.
- Jones, L. M., Fontanini, A., Sadacca, B. F., Miller, P., and Katz, D. B. (2007). Natural stimuli evoke dynamic sequences of states in sensory cortical ensembles. *Proc Natl Acad Sci U S A*, 104(47):18772–18777.
- Pearl, J. (2000). *Causality: models, reasoning and inference*. Cambridge University Press.
- Polani, D. (2006). Emergence, intrinsic structure of information, and agenthood. *Int J Complex Systems*, 1937.
- Seth, A. K. (2010). Measuring autonomy and emergence via Granger causality. *Artif Life*, 16(2):179–96.
- Shalizi, C. and Moore, C. (2006). What is a macrostate: Subjective observations and objective dynamics. <http://arxiv.org/abs/condmat/0303625>.
- Tononi, G. (2004). An information integration theory of consciousness. *BMC Neurosci*, 5:42.

# Walking, Hopping and Jumping: A Model of Transcription Factor Dynamics on DNA

David J. Barnes<sup>1</sup> and Dominique Chu<sup>1</sup>

<sup>1</sup>School of Computing, University of Kent, CT2 7NF, Canterbury, UK  
{d.j.barnes, d.f.chu}@kent.ac.uk

## Abstract

We present a model of how transcription factors scan DNA to find their specific binding sites. Following the classical work of Winter et al. (1981), our model assumes two modes of transcription factor dynamics. Adjacent moves, where the proteins make a single step movement to one side, or short walks where the transcription factors slide along the DNA several binding sites at a time. The purpose of this article is twofold. Firstly, we discuss how such a system can be efficiently modelled computationally. Secondly, we analyse how the mean first binding times of transcription factors to their specific time depends on key parameters of the system.

## Introduction

Regulation of gene activity can be understood as a computational process, in the sense that the cell reacts to changes in the environment by changing its internal states. There are several mechanisms the cell can use to make such internal changes. One important such mechanisms is the regulation of genes. In bacteria, gene regulation often involves the binding of regulatory proteins, so called *transcription factors* (TFs), to particular binding sites on the DNA.

One aspect that has commanded significant attention from bioscientists, physicists and systems biologists is the time required for regulatory proteins to find their target binding site on the genome. The problem is as follows: In order to turn a gene on (or indeed repress it) the TF needs to locate a specific binding site. The problem is that TFs are “sticky” to all parts of the DNA. When binding to the DNA a TF actually binds to an  $l$ -long sequence of nucleotides. The binding strength depends on the match between the bound sequence and an optimal pattern which represents the sequence of the specific binding site. The closer the match, the higher the affinity. While the binding affinity to specific sites is much higher than to most non-specific sites, the contribution of the latter is still significant enough to potentially “distract” a TF from locating its specific site. Furthermore, there are millions of non-specific sites and only few of the specific and active sites for each particular TF. Therefore, even though a TF spends very little time being bound to each of the non-specific sites, it may take a significant time to sample all

of them before the specific site is eventually found. The process of a TF finding its specific binding site necessarily limits the speed of a biological computation.

This problem, which has been known about for a long time, was first addressed by Winter et al. (1981), who proposed a random walk model of facilitated diffusion. The idea of this model is that the TF performs a mixed 1D and 3D random walk. The 1D random walk explores a small adjacent neighborhood of DNA, while the 3D random walk allows the TF to explore far-away, unconnected parts of the genome. It has been suggested by Wunderlich and Mirny (2008); Slutsky et al. (2004) and Murugan (2009) that the most efficient exploration of the genome, in the sense that it offers the fastest location of the specific binding site, is achieved when the 3D and 1D components are weighted approximately equally.

Most of the above work has been analytical. There are also a number of other results available. In this article we will describe an approach to building an efficient computer simulation model of TFs finding their specific binding sites (Barnes and Chu (2010)). This new approach will allow realistically sized simulations, thus significantly expanding the scope of previous models. The essence of the efficiency of the model is a careful management of memory to make the problem scalable, regardless of genome occupancy.

## The Model

The movement dynamics of TFs involves a search across a discrete (but very high) number of spatially organised binding sites. This suggests the potential for an individual agent-based modelling approach. The environment of the TF agents is a non-metric space; that is, there is no measure of distance between the agents. Embedded in this space is the DNA itself, which is represented as a string of the symbols  $a, c, g, t$  with periodic boundary condition. For all simulations reported here we used the genome of *E.coli* K12 (The University of Wisconsin (2009)). At any given time, every agent is either bound to one of the binding sites of the genome, or suspended in the non-metric space. We think of the space as a ‘reservoir’ of currently unbound TFs.

We define two types of agents, namely focal and non-focal TFs. We are primarily interested in the former, yet the latter are important in that their presence on the DNA could interfere with the search dynamics of the focal TFs. The number of non-focal TFs is kept constant during a specific simulation run (for reasons of computational efficiency), whereas the focal TFs are created and degraded with user-defined rates, hence particle numbers within the cell fluctuate over time.

Focal TFs have a definite binding *motif*  $\mathbf{m}$  that is used to determine their binding energy and, hence, their mean binding time at every DNA binding site in the model. If the length of the binding motif  $\mathbf{m}$  is  $l$  then the binding free energy to a particular sequence is calculated as follows:

$$F_s = \sum_{i=1}^l \omega_i \delta_{m_i, s_i} \quad (1)$$

Here,  $m_i$  is the  $i$ -th entry of the motif  $\mathbf{m}$ ,  $s_i$  the corresponding base of the actual binding sequence  $\mathbf{s}$  and  $\omega_i$  the empirically determined weighting factor of the binding motif. In contrast, non-focal TFs do not have specific binding sites; rather, they share low, position-dependent affinities to all sites on the DNA. Rather than calculating the binding energies dynamically, the affinity values for both types of TF are pre-calculated for every position on the DNA and stored in arrays of the same length as the DNA, making binding-time calculation very efficient.

The model update algorithm is event based, with three main classes of event available at each step:

- Create a focal TF.
- Bind a TF of either type to the DNA.
- Unbind a bound TF from the DNA.

Unbind events can result in complete unbinding into the reservoir or short, local 1D movements. Essential for the reliability of the model is to design the update algorithm so that the behaviour of the model is correct with respect to the choice of parameters (in the sense that it reproduces the statistics implied by the various binding and unbinding rates). To achieve this, we have adapted the Gillespie algorithm (Gillespie (1977)) to schedule events.

On every event, regardless of its class, only a single TF is updated. Breaking down the event classes in more details: an update consists of one of the following actions:

- A new focal TF is created and might attempt to bind.
- A TF binds from the reservoir to the DNA.
- A bound TF unbinds from the DNA into the reservoir.
- A bound TF moves to an adjacent binding site on the DNA.

- A bound TF makes a short move, i.e., binds with a uniform probability to an available binding site in the vicinity of its current site. The range of what counts as “vicinity” is user determined.
- A bound TF is destroyed.

## Scheduling of events

At model initialization all non-focal TFs are created and seeded onto random locations on the DNA via bind events at time zero. If there is insufficient space then the excess ends up in the reservoir. Then the creation times of all focal TFs are determined according to a user-defined rate, and creation events scheduled accordingly. Their lifetime is also determined at creation with a random number drawn from an exponential distribution with a mean of 1 over the deletion rate.

When its creation event occurs, a focal TF will immediately attempt to bind to a site on the genome with a user-defined probability; any such attempt is successful with a probability  $p = N_{\text{free}}/N_{\text{range}}$  where  $N_{\text{range}}$  is the total number of binding sites in range and  $N_{\text{free}}$  is the number of unoccupied sites in that range. We specify  $N_{\text{range}}$  because the initial bind attempt for a focal TF takes place within a limited, user-defined birth range on the DNA. This models the effect that (in bacterial cells) transcription and translation are performed in one step and hence proteins are produced close to their gene.

If the newly-created TF does not bind, then it is placed in the reservoir and may have the opportunity to attempt a general bind (i.e., one over the full range of the DNA) at a later time. The range restriction only applies to the initial binding attempt of a focal TF.

## Binding events

General binding is used both to seed initial occupancy of the DNA with non-focal TFs, and to support binding of both types of TF from the reservoir. A random available binding site is chosen from the full length of the DNA.

At the completion of every event, there is a probability that an unbound TF might attempt to bind from the reservoir. The time to the bind event is drawn from an exponential distribution with a mean of 1 over a value that depends upon the number of unbound TFs  $T_u$ , the number of available binding sites  $N_{\text{free}}$  along the full range of the DNA, and a constant factor  $k$ :

$$P(\text{bind}) = (k N_{\text{free}} T_u) \quad (2)$$

A new binding event will only be scheduled if it would occur before the next already scheduled event. This is because the binding probability depends on the current availability of binding sites which generally changes over time.

## Unbinding events

The duration time of a DNA-protein bond depends on the affinity of the type of TF for its binding site; specifically, for focal TFs this affinity is determined from equation 1. It is drawn from a Poisson distribution with mean  $\mu$ .

$$\mu = \exp\left(-\frac{F_s}{kT}\right)$$

Here  $k$  is the Boltzmann constant and  $T$  the absolute temperature. Binding from the reservoir onto the DNA is determined stochastically with a given user-determined rate.

At every unbind event, the next state of the TF is determined stochastically. Assuming that the TF has not reached the end of its life (in which case it would be destroyed), with a user-defined probability one of the following options will apply to it:

- the TF will attempt to make a one place move left or right (an immediately scheduled bind event);
- the TF will attempt a short move within a user-defined range either side of the previous binding site (an immediately scheduled bind event);
- the TF goes into the reservoir.

Either move could fail, due to roadblocks, and lead to the TF going into the reservoir. It should be clear from the above description that, on each iteration, the heart of the event loop is primarily concerned with: placing a TF on the DNA; removing a TF from the DNA; or both. Therefore, identifying free sections on the DNA is a potential performance bottleneck that could prevent scaling of the method to realistic sizes of both DNA and numbers of TFs.

## The memory model

The key to efficient implementation of binding and movement is the fast identification of available binding sites — i.e., not just empty bases but runs of bases that are at least as long as the binding motif (see eq 1) and can thus support binding of a TF. A naive representation of the DNA might be an array of Boolean values, one for each possible site, recording whether a site is currently occupied by a bound TF or not. In this implementation, an attempt to bind would involve the generation of a random number within the desired location range and a check as to whether that location is free or not. If it is not free then options might be: abandoning the attempt immediately; searching from that location in one or other direction until a free site is found; or identifying a fresh random location and repeating the process until a free site is found. While simple to implement, the weakness of this approach is immediately clear as the time to find a free location is dependent upon the occupancy of the DNA. Indeed, even when there are plenty of free individual bases, there are no guarantees that a long enough consecutive run

will exist to allow a TF to bind, and the approach outlined above must ensure that a search in vain will ultimately terminate.

Using this scheme the time to locate a free binding site depends on the occupancy of the DNA, and scales poorly with the size of the genome. In this model we therefore use a different approach that can find binding sites within a time independent of the occupancy. Rather than an unstructured array of Boolean status values we maintain a data structure that records all the remaining bindable sections of the DNA, as *(position, length)* pairs. The DNA is modeled as a 1D wrap-around structure. Note that because binding and unbinding occur at irregular intervals, sections of binding sites are occupied and freed according to no particular regular pattern. The resulting space management problem is akin to *dynamic storage allocation* in program runtime environments (Knuth (1997)), as opposed to *stack* (last-in, first-out) memory management, for instance. A significant difference, however, is that traditional allocation algorithms, such as *first fit* and *best fit*, are inapplicable in this context, because the memory manager must always allocate a particular section of free space that has been selected by the bind event, rather than having a free choice. In common with dynamic memory allocation, available space quickly becomes “fragmented”. For instance, consider a run of  $l + n$  unoccupied sites, where  $l$  is the length of a TF to be bound and  $n \geq l$  (Figure 2a). This sequence offers  $n + 1$  potential binding sites before a bind but anywhere between 0 and  $n - l + 1$  sites after the bind, depending on where the bind takes place within the run and the size of  $n$  in comparison to  $l$ . If the TF were to bind across the middle of the section then the two fragments either side may well be too short to support another TF (Figure 2b). As a result, the data structure recording bindable sections must be supplemented by a similar data structure recording unbindable fragments. For both we use the `set` associative container from the C++ STL (Meyers (2001)), which provides efficient access via its key which, in our case, is the binding position. Note that a fragment resulting from the bind of one TF may become usable before that particular TF unbinds — as a result of the earlier bound TF occupying the adjacent section at the other end of the fragment becoming unbound (Figure 2c). Indeed, most of the complexity of the memory management occurs during the bind-unbind cycle, at the point where a TF unbinds and the section it occupies becomes available again. Before being returned to the set of available sections it must be reunited with any fragments at either end. In addition, the newly freed section may now be contiguous with another already available section, in which case the two must be coalesced into one.

## Methods

All simulations in this article were performed by starting with an empty wraparound DNA of length 4639675 at time

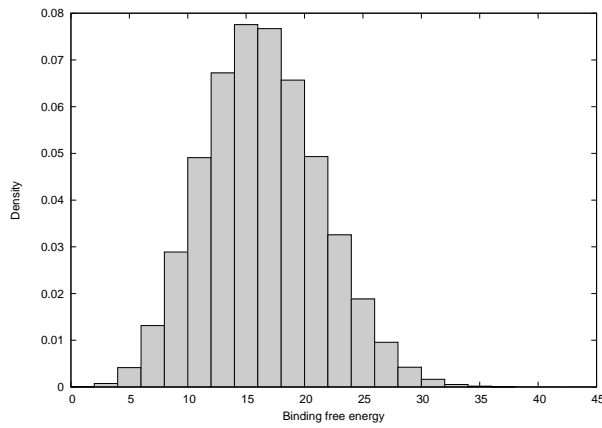


Figure 1: A histogram of the binding free energies as calculated from eq. 1. The energies are Gaussian distributed.

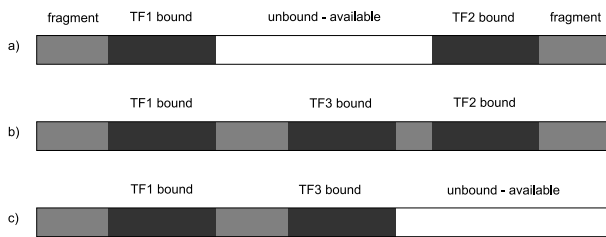


Figure 2: DNA section illustrating fragmentation and de-fragmentation during TF binding and unbinding; a) Two bound TFs, fragments and an available section; b) A third TF binds, resulting in two new fragments; c) A TF unbinds, fragments become available again.

0. Upon starting, the simulation protein was created with a rate of 0.01. The degradation rate of protein was 0.0009. Each simulation was run for a maximum of  $10^9$  time units, but was halted as soon as a TF was bound to the specific binding site at position 4540692 on the DNA. The halting time was taken as the mean first binding time (MFBT) referred to below. For each set of parameters the MFBT was calculated from 10 independent simulations (unless specified otherwise). In the graphs below, each point indicates the MFBT where the mean has been taken over the set of simulations that had been performed. Error-bars and standard deviation are not indicated in the graphs to preserve legibility. In nearly all experiments we performed, the standard deviation is comparable to the mean, indicating that typical binding times deviate significantly from the mean.

The source code of the program used here is available for free download.<sup>1</sup>

<sup>1</sup>via anonymous FTP from [ftp.cs.kent.ac.uk](ftp://ftp.cs.kent.ac.uk/pub/djb/exp/exp-distrib.tgz) as [pub/djb/exp/exp-distrib.tgz](ftp://ftp.cs.kent.ac.uk/pub/djb/exp/exp-distrib.tgz)

## Results

One of the main variables to consider is the time the TF requires to reach its specific site. For a single random walker it is expected that MFBT scales with the square root of the distance. In the case of an ensemble walking this may be different. We decided to check this. To this end we performed a number of experiments with the following setup: We chose a synthesis site at which the TFs were produced. This has the effect that the TFs would attach at random to the binding site within a specified window. This introduces a stochastic element into the simulation, in the sense that not all TFs start from the same site. Some will start closer to the specific site, some from farther away. This choice has another effect. It limits the number of TFs that can attach to the DNA per time unit. The reason is that, upon binding to the DNA, TFs either occupy the binding sites within the initial binding window or they are released into the cytoplasm (represented by the “reservoir” in our model). If all sites within this window are occupied, no further TFs can bind and newly synthesised TF will always be released into the cytoplasm. We set the parameters such that no binding from the cytoplasm to the DNA is possible; hence, for the purpose of our simulation, once a TF unbinds from the DNA it is, in effect, lost forever. We found that the initial binding window is a strong restriction on the number of bound TFs.

We first performed a number of simulations with the initial binding window equal in size to the DNA. The effect of this is that newly created TFs will bind anywhere on the DNA. We allow the TFs to perform short moves of length up to 50 binding sites at a time; adjacent moves happen with a probability of 0. In this case we would predict that the MFBT is independent of the location of the synthesis site, but we would expect that the MFBT decreases as the TFs can travel faster, that is a higher short move length should lead to lower MFBTs. We varied both the probability of short moves and the site where TFs are synthesized. Figure 3 summarises the results of these simulations and confirms that the synthesis site is irrelevant, as expected. The graph shows the MFBT when all movements are only adjacent neighbor moves ( $P = 1$ ), they are all short move events ( $P = 0$ ) and an in-between case ( $P = 0.8$ ). For other values of  $P$  we found that the MFBT always increases with increasing  $P$ . As can be seen from figure 3 the difference between the MFBTs for extreme cases of  $P$  are at the order of a magnitude.

In bacteria translation and transcription are closely inter-linked. This means that protein tends to be made in close spatial proximity to the gene that codes for the particular protein. Following gene synthesis there is thus an increased chance that a TF binds to a particular local region of the DNA. We investigate the effect of this on the MFBT by varying the location of the initial binding window. Figure 4 shows a number of simulations with a window size of 40 (20 on each side of an assumed protein synthesis site). Such a

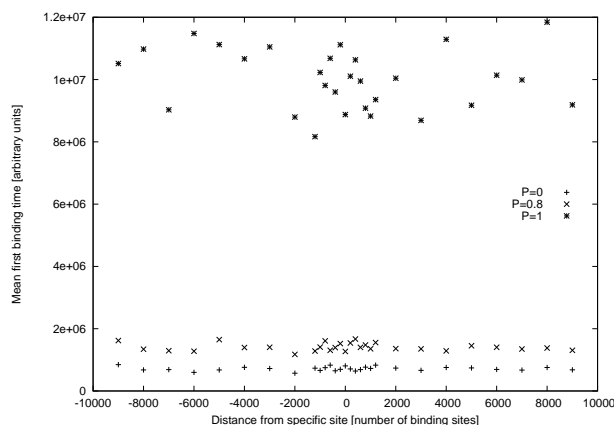


Figure 3: The mean first binding time to reach a particular specific site as a function of the short move distance. The window size equals the size entire genome. The short move length was set to 50 in these simulations.

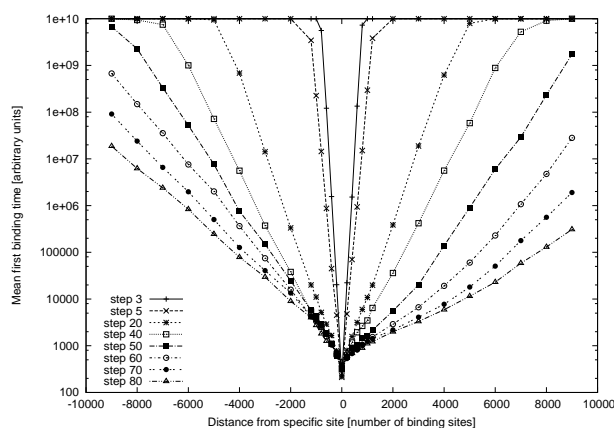


Figure 4: The mean first binding time to reach a particular specific site as a function of the short move distance. The short move probability was set to 0.9.

small preferred binding window is, admittedly, biologically unrealistic. However, it was chosen for practical considerations relating to the simulation speed. We found a strong dependence of the MFBT on the protein synthesis location as summarised in figure 4.

From these experiments it seems that a higher short move probability speeds up the search process. However, we would expect that the importance of this effect depends on the proximity of the synthesis site to the specific binding site. If the binding site is very close to the synthesis site, then one would conjecture that large step sizes will tend to “overshoot,” that is they will simply miss the specific site during the movement. With larger initial distances this overshoot will happen as well, but TFs will move faster into the proximity of the specific site, hence counteracting this effect.

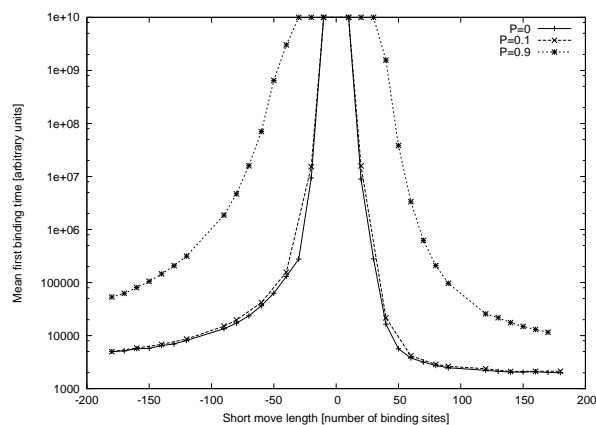


Figure 5: The mean first binding time to reach a particular specific site as a function of the short move distance. The short move probability was set to 0.9.

We performed a variant of the above experiments to understand this in more detail. The graph in figure 5 shows simulations where we kept the initial binding site fixed at an offset of  $\pm 3000$  binding sites from the specific site. The  $x$ -axis shows the short move length in the simulation and the sign of the  $x$ -axis indicates the centre of the initial binding site. So, for example the point marked at  $x = -100$  represents a simulation with an offset of the initial binding site of  $-3000$  from the specific binding site, and a short move length of 100. In these simulations each point represents the average MFBT over 1000 simulation experiments. The graph shows values for 3 different adjacent move probabilities, corresponding to all movements are short-moves, 90% of all events are short-move events and 10% of all move events are short move events.

The graph is somewhat complex to interpret, but shows that the MFBT falls faster than exponential with the short move length. For  $P = 0$  and  $P = 0.9$  the MFBT decreases by several orders of magnitude as the short move length increases from 20 to 100. When the short move length is smaller than 20, then irrespective of the value of  $P$  in the simulations shown here the MFBT is larger than the maximum simulation time of  $10^9$  time units.

A closer look at the simulation results, particularly at figure 5 reveals that the MFBT is asymmetric around the specific binding site. When the binding site is to the right of the specific site (i.e., higher id-numbers in the coordinate system used here), then the MFBT tends to be lower than when the TF is synthesised to the left. This effect is clearly illustrated in figure 5. Particularly for high short move length values there is a clear difference between the two synthesis sites. For example, when the short move length is 180, then for the parameters used in the figure the difference between the MFBTs amounts to nearly a factor of 2.

The underlying cause of the difference appears to be the



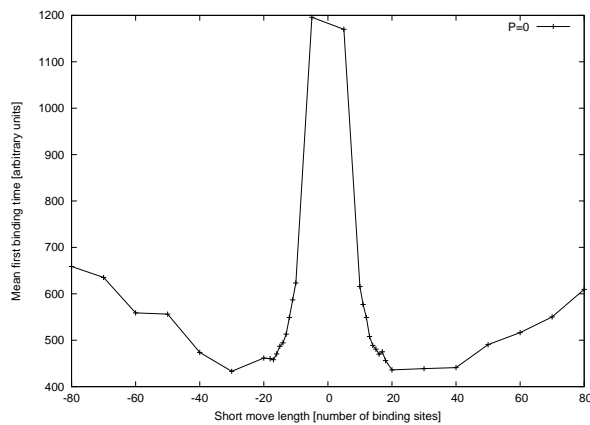


Figure 6: Same as figure 5, but with  $P = 0$  and with an offset of  $\pm 100$ .

presence of 3 further specific binding sites to the right of the focal site we are interested in. These three additional specific binding sites are in very close spatial proximity to the focal site with offsets of 18, 303 and 321 binding sites respectively. One can think of the dynamics as follows: When a TF binds to one of the 4 binding sites, then it acts as a reflecting boundary for random walkers in the area, confining random walkers within the area of the specific binding sites. This has the net effect of reducing the MFBT for the random walkers.

In figure 5 it appears that the longer the sliding distance, the shorter the MFBT. This is somewhat counter-intuitive. We would expect that there is an optimum sliding distance, which allows fast approach of the specific binding site, while balancing this with the problem of over-shooting the specific site. Within the short move distances considered in figure 5 such an optimum is not apparent. However, we would expect that such an optimum short move distance depends on the distance of the synthesis site from the specific site; the closer the synthesis site, the shorter the optimal short move distance. To check this we performed another set of experiments varying the short move distance, but with synthesis sites located at an offset of  $\pm 100$ . Figure 6 shows the results. It is apparent that there is a clear minimum MFBT for both offsets, as expected.

## Discussion and Conclusion

In this contribution we have presented a model that supports the efficient simulation of the process of TFs finding their specific binding sites. One of the problems that we had identified was that realistic simulations are computationally extremely demanding. For this reason, modeling of specific binding site localisation has been restricted to mathematically tractable but unrealistic models. Here we have made the first steps towards a computationally feasible implementation. One of the bottlenecks we have identified is the lo-

calisation of free binding sites on the DNA. By adapting approaches from dynamic memory allocation we were able to achieve speedups with respect to a naive algorithm of many orders of magnitude.

Apart from finding an efficient simulation implementation, we found that the MFBT depends in a complicated way on the short move distance, the synthesis site, but also the local configuration of the binding sites. Our simulations are a significant extension (although in simulation) to the analytical results developed by both Murugan and Mirny *et al.* The picture emerging from these simulations is that the situation is significantly more involved than suggested by these previous articles. For example: One of the conclusions by Murugan was that there is an optimal division between adjacent moves and short moves. We could not reproduce this in our setup. Instead we found that, up to the range we investigated, short moves are generally faster and more efficient than adjacent moves. We do not mean to imply that their conclusions are wrong. However, it is clear that the conclusions of various models are not robust with respect to variations of underlying assumptions. This is normally a worrying sign in modelling.

This suggests that a more thorough investigation of this system is necessary, in order to come to a clear understanding of how previous mathematical results relate to the simulation results obtained here.

## References

- Barnes, D. and Chu, D. (2010). An efficient model for investigating specific site binding of transcription factors. In *Proceedings of the 4th International Conference on Bioinformatics and Biomedical Engineering, June 18-20, Chengdu, China, 2010*, page 4. IEE Xplore.
- Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81(25):2340–2361.
- Knuth, D. E. (1997). *Art of Computer Programming, Volume 1: Fundamental Algorithms (3rd Edition)*. Addison-Wesley Professional.
- Meyers, S. (2001). *Effective STL: 50 Specific Ways to Improve the Use of the Standard Template Library*. Addison-Wesley Professional Computing Series. Addison Wesley.
- Murugan, R. (2009). Packaging effects on site-specific dna-protein interactions. *Physical Review E*, 79(6 Pt 1):061920.
- Slutsky, M., Kardar, M., and Mirny, L. (2004). Diffusion in correlated random potentials, with applications to DNA. *Physical Review E*, 69(6):061903.
- The University of Wisconsin, M. (2009). *E. coli genome project*.
- Winter, R., Berg, O., and von Hippel, P. (1981). Diffusion-driven mechanisms of protein translocation on nucleic acids. 3. the *escherichia coli lac* repressor-operator interaction: kinetic measurements and conclusions. *Biochemistry*, 20(24):6961–6977.
- Wunderlich, Z. and Mirny, L. (2008). Spatial effects on the speed and reliability of protein-DNA search. *Nucleic Acids Research*, 36(11):3570–3578.

# Coming Phase to Phase with Surfactants

Stuart Bartlett<sup>1</sup> and Seth Bullock<sup>1</sup>

<sup>1</sup>Institute for Complex Systems Simulation, University of Southampton, Southampton, SO17 1BJ  
S.J.Bartlett@soton.ac.uk

## Abstract

We introduce a fast cellular automata model for the simulation of surfactant dynamics based on a previous model by Ono and Ikegami (2001). Here, individual lipid-like particles undergo stochastic movement and rotation on a two-dimensional lattice in response to potential energy gradients. The particles are endowed with an internal structure that reflects their amphiphilic character. Their head groups are weakly repelled by water whereas their hydrophobic tails cannot be readily hydrated. This leads to the formation of a variety of structures when the particles are placed in solution. The model in its current form compels a myriad of potential self-organisation experiments. Heterogeneous boundary conditions, chemical interactions and an arbitrary diversity of particles can easily be modelled. Our main objective was to establish a computational platform for investigating how mechanisms of lipid homeostasis might evolve among populations of protocells.

## Introduction

The debate concerning the containers within which the first biochemistries developed is hotly contested. One uncontroversial observation however, is that Nature has since fixed upon a single class of molecule to use as the barrier between cell contents and the external environment, be it the intercellular space or the outside world. These special molecules - lipids - possess the crucial property of being amphiphilic, they contain both hydrophilic and hydrophobic groups. Amphiphilic molecules are an example of a surfactant, a substance which reduces the interfacial tension between two fluids (we shall use the terms amphiphile, surfactant and lipid interchangeably). They are thus endowed with an ability to arrange themselves into meso-scale structures when placed in solution. The shapes of these structures reflect the systems' attempts to minimise contact between hydrophobic groups within the amphiphiles, and water molecules. One such structure, the vesicle, can be conveniently used to separate one aqueous environment from the surrounding water. It is this molecular device that organisms have adopted as a means of separating the inner cell space from its exterior. All nutrients and waste products must pass through this barrier in order to carry out their function within a cell. The

membrane must also grow, sever, re-connect and undergo various other transformations during the cell cycle. Other lipid membranes separating organelles from the intracellular space must also adopt various shapes and curvatures in order to maximise their function. Given the tremendous importance of membranes both during the early stages of the evolution of life and in contemporary organisms, it is easy to justify the pursuit of a complete understanding of amphiphile dynamics.

The electrostatic interactions between the constituents of lipid molecules are fairly well understood as are the equations of motion for the behaviour of such molecules in solution. Furthermore, the equilibrium properties of surfactant-water-oil systems have been analysed using a lattice model first introduced by Widom (1986). Having successfully reproduced some key features of surfactant phase diagrams, Widom's simple lattice model as well as other spin-based models (so-called due to their being isomorphic to a spin- $\frac{1}{2}$  Ising model), stimulated a profusion of investigations to be carried out both analytically and through Monte Carlo simulation (Larson et al., 1985). For a summary of the research performed during this period, see also the review of Kawakatsu et al. (1994). Despite these successes, there remain significant analytical obstacles to the complete understanding of more complex, biologically relevant lipid systems (for example, if we wish to include processes such as the synthesis and decay of lipids through metabolic pathways). Life is the antithesis of thermodynamic equilibrium, and exhibits highly non-linear dynamical behaviour to boot. These factors, among others, have presented what appear to be insurmountable barriers to a pure analytical understanding of the higher level systems of molecular biology. Instead we must, for the time being at least, look to computational methods. Even numerical integration of the equations of motion is a daunting task. Real systems of interest involve massive numbers of molecules, and interesting dynamic behaviour occurs over time scales which are much longer than typical numerical integration steps. Therefore models derived from first principles which solve the exact system of equations (molecular dynamics) are very expensive in terms

of computational resources. Multi-scale and hybrid models have been put forward by Ayton and Voth (2002) and Lyubartsev (2005) for example, but simulations over longer time scales and mesoscopic length scales with the potential for variable environments and boundary conditions are still relatively rare.

The popularisation of cellular automata (CA) models has given birth to a family of simulation techniques which have shown considerable promise for modelling complex systems such as amphiphile solutions (Kier et al., 1999; Nilsson and Rasmussen, 2003; Rothman and Zaleski, 2004). CAs are discrete time and space models in which all interactions occur on a local scale. These properties allow CAs to be much less computationally demanding than traditional numerical schemes. The lattice and discretisation constraints of CAs can cause problems with respect to invariance under geometric transformations but there is one class which has been shown to mimic reality with surprising effectiveness. So-called lattice gas models simulate hydrodynamics by allowing a set of particles to move and collide on a lattice. The rules of interaction are defined such that mass and momentum are conserved and one can derive the Navier-Stokes equations from the microdynamical rules of the CA (Frisch et al., 1986). The basic lattice gas has been extended for a variety of applications including the fluid dynamics of water-oil-surfactant mixtures (Boghossian et al., 1996, 2000; Mayer et al., 1997). Both of these models were successful in re-creating some key lipid phases and were later applied to more specific systems including, in the case of the model of Boghossian et al. (1996), self-reproducing micelles (Coveney et al., 1996), which showed impressive agreement with the experimental results of Bachmann et al. (1992).

In this investigation, we explored the abilities of a new CA for the simulation of amphiphile solution systems based upon the artificial chemistry model of Ono and Ikegami (2001). This model differs from the lattice gases mentioned above. Particles move in pursuit of potential energy minima, but they do not collide and exchange momentum. We perform this simplification of neglecting the individual particle momenta because we wish to focus on the self-assembly process and the meso-level dynamics of more complex systems with variable boundary conditions. By ignoring the explicit hydrodynamics of the system, the formulation of the model is greatly simplified as are the computational demands. We believe that the key dynamics of the self-assembly process are nevertheless retained.

An important feature of our simulations is the way in which surfactants are defined. Rather than a generic ‘membrane’ particle, we have applied a more explicit representation of the internal structure of amphiphiles. In addition, we introduced three different lipid species, each with its own geometry. Real cell membranes consist of many different lipid types. Some of them naturally form bilayers but there are also non-bilayer forming lipids present. The exact func-

tion of these non-bilayer lipids has been debated for many years and it is likely that they play several roles in the cellular performance (Lindblom et al., 1986). The stability, robustness and versatility of cell membranes derives in part from the homeostatic balance of the distribution of these varied lipid species (Beard et al., 2008). Therefore we aimed to endow our model with the additional freedom of having lipids with a range of membrane-forming properties. Our objective was to construct a platform which we could use to investigate the spontaneous evolution of lipid homeostatic mechanisms. Ono and Ikegami (2001) have already shown that simple cell-like entities arise spontaneously within their model framework. We aim to extend that model framework such that we can simulate not only the formation of protocells, but also the evolution by those protocells of mechanisms for balancing the lipid composition of their membranes. In our model, the geometry of vesicles (or protocell membranes) resulting from the spontaneous organisation process depends not only on environmental factors but also on the distribution of the different lipid species, since each species has its own preferred membrane curvature.

In this paper we wish to present the model in its current form as a tool for simulating an interesting and important class of complex system. As well as simulating the emergence of lipid homeostasis, the model could easily be modified to simulate complex reaction-diffusion or self-reproducing micellar systems, among others. We shall first give a brief description of the workings of the model, before describing the main results of our investigations so far. These will include simple phase separation of water and hydrophobic monomers, micelle formation, bilayer formation, ternary mixtures leading to monolayer formation and finally a set of hysteresis experiments. We shall then conclude with a discussion of the significance of these results before suggesting some relevant systems which will be simulated by our model in the future. Due to space restrictions, we shall not present a strong focus on the technical details of the model, instead we shall describe the most important features and highlight its phenomenological successes.

## Model Description

The mechanics of our model are essentially the same as those of Ono and Ikegami (2001); Ono (2005). The simulation domain is a 2-dimensional triangular lattice over which particles move and interact. An arbitrary number of particles can reside on each lattice site, and the boundaries of the lattice are periodic. The model proceeds via a standard metropolis algorithm (relaxation towards a global potential energy minimum). All interactions between possible particle pairs across all relative orientations are defined *a priori* in the form of a lookup table. Same site interactions consist of a strong excluded volume repulsion which is the same for all particle types and acts between all particle types. The nearest-neighbour interactions take several different forms.

All forces are repulsive but the strength depends on the physical properties of the two particles involved in the interaction. In order to approximate the effect of hydrogen bonding, water particles repel one another with almost negligible force. Hydrophobic monomers also repel each other weakly. There is a strong repulsion felt between water particles and hydrophobic monomers. This is due to the frustration of the surrounding water molecules, which are unable to satisfy all of their potential hydrogen bonds. Interactions involving surfactants are slightly more complicated.

The crucial differences between our version of the model and that of Ono and Ikegami (2001), are the structure and interactions of the surfactants. We make use of a more explicit representation of lipid particle geometry. Although the surfactants have internal structure, we do not model the harmonic motion of the individual molecular components. Each surfactant is represented as a rigid particle free only to rotate in discrete increments (reflecting the discrete nature and underlying symmetry of the lattice). The pairwise interactions between these particles are computed using the sum of a set of Lennard-Jones functions. These calculate Van der Waals forces for the four interactions between the hydrophilic heads and hydrophobic tails of all pairs of surfactants which are nearest neighbours. Physically, these four terms represent the dipole-dipole interaction between the polar head regions, the dipole-induced dipole interactions between the heads and hydrocarbon tails, and the induced dipole-induced dipole interaction between the two tail regions.

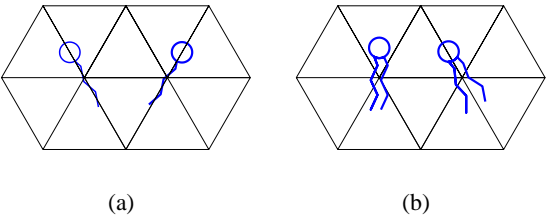


Figure 1: Equilibrium orientations of pairs of surfactant particles. a)  $M_1$  particles at adjacent lattice sites align with their tails closer than their heads due to their cone-like geometry. b) an  $M_2$  particle and an  $M_3$  particle prefer to align with an angle of  $\pi/6$  between their axes. This is due to the wider splay of the tails of  $M_3$  particles.

$M_1$  particles are modelled on detergent particles with single alkyl chains. This gives them a cone-like shape with a broader head region. Figure 1(a) shows a schematic illustration of their pairwise equilibrium configuration. The two  $M_1$  particles align with an angle of  $\frac{\pi}{3}$  between their vertical axes.  $M_2$  particles are based on lipids with double hydrocarbon chains giving them a cylindrical geometry. As a result

they prefer to align parallel with one another.  $M_3$  particles have broader tail regions, wider than their head groups. A second example of the equilibrium configuration of a pair of particles is shown in figure 1(b), which illustrates how the cylindrical  $M_2$  particle and the broad-tailed  $M_3$  particle prefer to align with one another. Since the  $M_2$  particle has a cylindrical geometry but the  $M_3$  particle has a broader tail region, these two particles prefer to align with an angle of  $\frac{\pi}{6}$ . The other equilibrium configurations are defined in a similar way, e.g. an angle of  $\frac{\pi}{3}$  for pairs of  $M_3$  particles (with head groups closer than tails) and an angle of 0 for pairs consisting of an  $M_1$  and an  $M_3$  particle.

We now turn to defining the interactions between surfactants and water. Clearly the head groups of  $M_1$  particles will be attracted to water over a broader range of angles than those of  $M_2$  and  $M_3$ . The repulsion between the tails of  $M_3$  particles and water will also extend over a wider range of angles than the other two particles. These varying affinities for water are summarised in figure 2, which shows the variation of the pairwise potential  $\phi$  for an amphiphile neighbouring a water particle over a range of orientation angles  $\theta$ . The  $M_2$  particle with its cylindrical geometry, feels an anti-symmetric repulsion as a function of  $\theta$ . Conversely, the  $M_3$  particles feel a broad ranged repulsion when their tails face water and only over a narrow range do they experience an attraction to water.

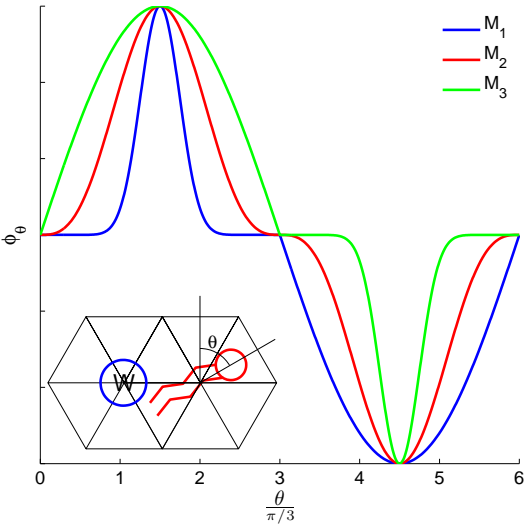


Figure 2: Pairwise potential  $\phi$  for a water and surfactant particle at neighbouring lattice sites. The potential varies as a function of the surfactant orientation  $\theta$  and takes on a different functional form for each of the three surfactant species. Note that in the model, the possible values of  $\theta$  are discretised, the continuous curves are indicative only.

At each time step, the potential energy field for each particle type is calculated using the interactions described above.

Particles then undergo stochastic transitions in pursuit of local energy minima. These transitions consist of translation by one lattice spacing and rotation (in the case of the anisotropic surfactant particles). Particle states are updated synchronously. As the particles relax into local minima the system as a whole tries to reach a state of global energy minimum analogous to the process of simulated annealing. The probability of a particle undergoing a transition is proportional to the value of the energy response function, evaluated for that transition (Ono and Ikegami, 2001):

$$f(\Delta\Phi) = \frac{\Delta\Phi}{e^{\beta\Delta\Phi} - 1} \quad (1)$$

$\Delta\Phi$  is the potential energy change of the transition and  $\beta = 1/T$  is the inverse temperature (we take Boltzmann's constant equal to unity). This function is designed to implement the basic character of a Boltzmann factor without the risk of the value diverging for large negative  $\Delta\Phi$  (transitions which are energy-reducing are not guaranteed to be accepted, as in a standard Monte Carlo algorithm). Simulations proceed by making use of this function to calculate transition probabilities. Particle states are then updated synchronously and randomly, biased by these probabilities.

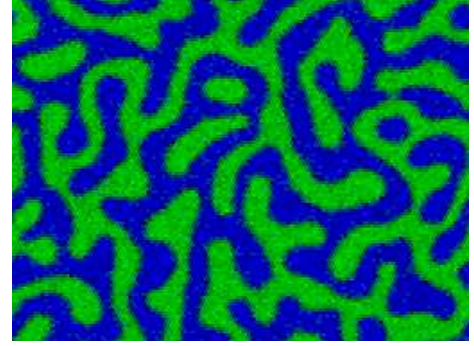
## Results

We shall now examine the most important results from testing the model over a range of conditions. Note that all images shown here are sections taken from larger systems therefore the boundaries in the images do not wrap around in the periodic way that they do in the simulation. In all figures, depth of green corresponds to the concentration of oil particles, depth of blue corresponds to water concentration and depth of red to surfactant concentration.

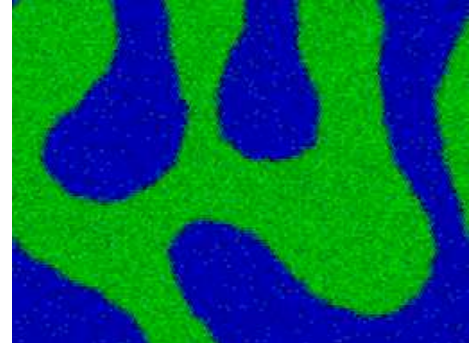
### Phase Separation

We begin with a simple, characteristic situation: a 50:50 water-oil mixture. Since polar and organic solvents do not mix due to their differing capabilities for hydrogen bond formation, we would expect such a system to relax to a state of phase separation in which the surface tension (interfacial area or contour in 2D) between the two substances tends to a minimum. Figure 3 shows two snapshots from a simulation containing average densities of  $\rho_w = \rho_o = 7.5$  particles per lattice site, at a temperature of  $T = 0.8$ . We can see in figure 3(a) that the system separates into regions of almost pure water and oil in the early stages. Over time the average curvature of the interface between the two regions is persistently reduced. Close observation reveals the propagation of capillary waves across this interface (see animation at: <http://tinyurl.com/lipid-CAs>), a characteristic surface tension effect. Given sufficient time, the system will reach a state of a single straight interface separating the two phase regions. Since the relaxation time scales approximately exponentially with system size, one would have to

run the simulation for an extremely long time to reach this state. However we can already see this minimum energy configuration at smaller length scales within the system.



(a)



(b)

Figure 3: System configuration after a)  $t = 1 \times 10^4$  and b)  $t = 1 \times 10^6$  time steps for a binary mixture of water and oil particles.

### Surfactant-Water Mixtures

We now turn our focus to the behaviour of the surfactant particles in the presence of water. Experimental results from studies of real lipid systems lead us to expect structures such as micelles, bilayers and vesicles among others (Tresset, 2009). We also know that the appearance of such structures should depend on certain parameters such as the temperature and surfactant concentration.

**Micelles**  $M_1$  particles were designed to emulate detergent molecules with single alkyl chains. We represented this in the model by endowing them with a cone-like structure: a narrow tail region and broad head section. We would expect such particles to coalesce into micelles in the presence of water. In a micellar configuration, the contact between hydrocarbon tails and water is minimised whilst the energetic aspirations of the amphiphiles are also reasonably satisfied. We can see from figure 4 that the equilibrium structure exhibited by  $M_1$  particles is the micelle. The configuration

shown contains average densities of  $\rho_w = 12$  and  $\rho_{M_1} = 3$  particles per lattice site and had a temperature of  $T = 0.3$ . Increasing the surfactant concentration does not alter the micellar configuration, it simply causes a greater number and hence closer packing of micelles. Likewise decreasing the  $M_1$  concentration simply results in a smaller number of micelles as the surfactants have a lower probability of encountering one another as they perform random walks over the lattice. Given time they do begin to aggregate but the process is slow. Furthermore, if the temperature is high, the micelles cannot form because they require a certain threshold number of constituents before they can remain robust to thermal fluctuations. Micelles containing only a small number of particles are not robust to these perturbations and thus do not persist. Hence at low surfactant concentrations, micelles can only form at low temperatures where fluctuations are less frequent. Below the critical micelle concentration (CMC), micelles would be unable to form at any temperature. In our model the CMC is very low ( $\rho_{M_1C} < 1$  particle per lattice site) and we have not yet explored such low surfactant concentrations.

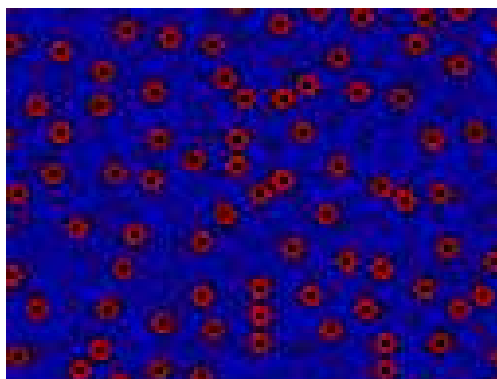


Figure 4: System configuration after  $t_f = 2 \times 10^5$  time steps for a binary mixture of water and  $M_1$  surfactant particles.

**Bilayers** We defined  $M_2$  particles as being similar to lipids with a cylindrical geometry. Their energetic requirements are satisfied if they align parallel with one another forming a straight bilayer. As was the case for  $M_1$  particles, this arrangement minimises the interfacial contact between hydrophobic tails and water while also satisfying the energetic preferences of the surfactants. Figure 5 shows a typical steady state of a water- $M_2$  mixture. It is clear that the self-assembly properties of this surfactant species are quite different from those of the  $M_1$  particle. Under identical conditions and concentrations,  $M_2$  particles assemble into bilayers, in contrast to the micelles formed by the  $M_1$  particle. If the concentration of  $M_2$  particles is very low, i.e.,  $\rho_{M_2} < 1$  particle per lattice site, below the critical bilayer concentration (CBC), micelles are formed rather than bilay-

ers. However they do not possess the central voids of the  $M_1$  micelles so they could also be described as small clusters. At these low concentrations, tuning the temperature to a critical value of  $T \approx 0.3$  allows a small number of bilayer sections to form but they are rapidly destroyed once the temperature reaches  $T = 0.4$ . This critical structure formation is analogous to the formation of micelles at low concentrations described in the previous section. Further investigations will reveal the nature of this transition region, within which well-defined structures form, but outside of which no such structures persist. As the concentration of  $M_2$  particles is increased above the CBC, the system becomes more densely packed with bilayers and the interconnectivity of the bilayers increases concomitantly.

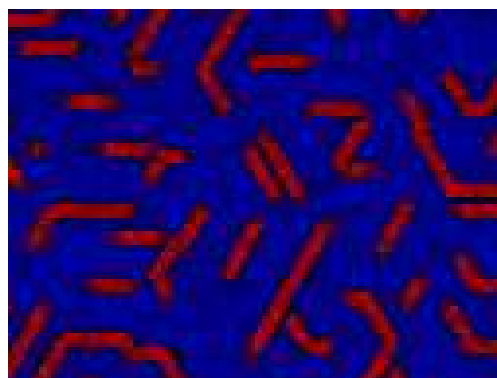
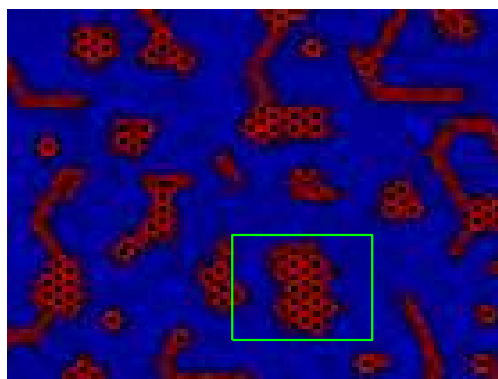


Figure 5: System configuration after  $t_f = 2 \times 10^5$  time steps for a binary mixture of water and  $M_2$  surfactant particles.

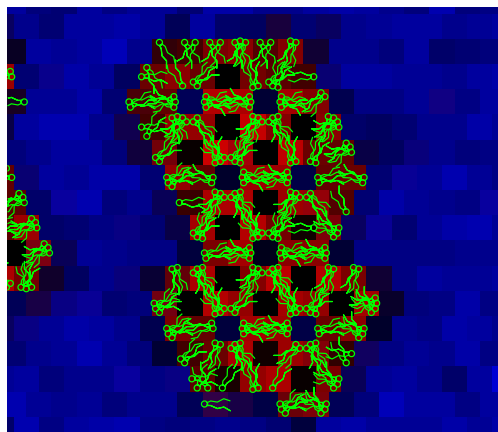
**Bilayers and Reverse Micelles** The  $M_3$  surfactant possesses a broad tail region so it should be averse to micelle formation. Bilayers are also not the ideal structure since pairs of  $M_3$  particles would prefer to align with an angle of  $\frac{\pi}{3}$  between their long axes. In an organic solvent these particles would form reverse micelles, but it is not obvious what structures they would form in a polar solvent. Because  $M_3$  particles prefer not to form micelles or bilayers, they actually attempt to create an environment in which reverse micelle formation is possible. Figure 6 shows the equilibrium configuration of a mixture of water and  $M_3$  surfactants at the same concentration and temperature as the systems shown in figures 4 and 5. We can see that the system adopts a mixture of bilayers and clusters. On closer inspection, one finds that the clusters consist of amphiphiles forming a hexagonal phase. There are water particles at the centres of the reverse micelles due to the surfactant head groups being water soluble. In contrast the inter-micellar voids are just that, they are devoid of particles since they are apolar environments.

**Monolayers** Having evaluated the behaviour of water-oil and water-surfactant mixtures, we can now explore the equi-





(a)



(b)

Figure 6: a) System configuration after  $t_f = 2 \times 10^5$  time steps for a binary mixture of water and  $M_3$  surfactant particles. b) A closer view of the highlighted region from the upper figure showing a cluster of  $M_3$  particles which have assembled into a honeycomb structure allowing the formation of reverse micelles. Particles are drawn where there are 2 or more surfactants present in that position and orientation.

librium configurations of ternary solutions. The polar and organic solvents should again separate but now the surfactants can take up positions along the phase boundary in order to further minimise the total surface free energy. The surfactants should align themselves such that their polar heads are hydrated and their lyophilic tails mingle with the oil regions. Figure 7 displays such behaviour when we initialise a simulation with average densities of  $\rho_w = \rho_o = 7$  and  $\rho_{M_2} = 1$  and allow it to relax for  $t_f = 2 \times 10^5$  time steps at a temperature of  $T = 0.1$ . The surfactants rapidly line the oil-water interface and at low temperatures the system reaches a steady state where the oil islands become stationary and almost completely cease to merge or divide. At higher temperatures the system adopts a configuration identical in geometric character to that in figure 8(a). At this temperature,  $T = 0.4$  the system has more freedom to explore its mi-

crostates and hence over time the oil regions merge, grow in size and change shape in an effort to minimise their average curvature, analogous to the situation for the binary oil-water case. Fluctuations present in the initial conditions are gradually damped out. At these high temperatures, the process of potential energy minimisation struggles since the energy response function makes less of a distinction between transitions which are energy-reducing and those which incur an energy cost. So although there is phase separation and the surfactants assemble on the phase boundary, there are also surfactants spread thinly across the entire lattice. The bulk phase separation effects dominate here due to the large numbers of water and oil particles present and the high temperature. In contrast, at lower temperatures, the presence of the surfactants is more influential. This is visible in figure 7. Since  $M_2$  particles prefer to align parallel to one another, the oil-water interface takes on a slightly different appearance. It is composed of straight sections punctuated by sharp corners, typically turning through angles of  $\frac{\pi}{3}$ . Because the surfactant monolayer is rather inflexible at this low temperature, the system does not undergo any significant geometric changes once settled into the bicontinuous state shown in figure 7. We also explored situations in which the average oil densities were lower than the water densities. In these cases, the so-called microemulsion phase is exhibited, in which droplets of oil form, surrounded by surfactant boundaries. These droplets were seen to merge when they encountered one another.

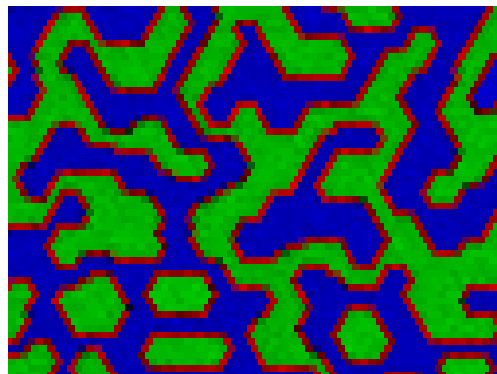
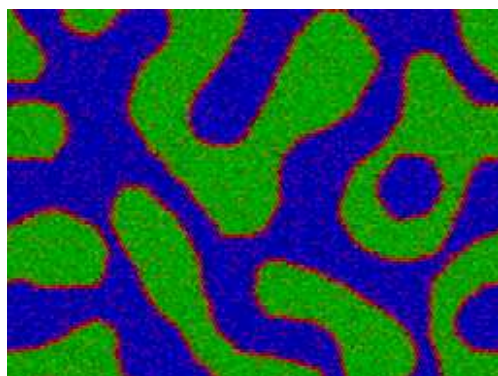


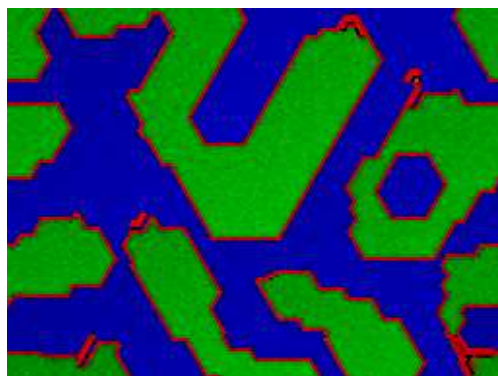
Figure 7: System configuration after  $t_f = 2 \times 10^5$  time steps for a ternary mixture of water, oil and  $M_2$  surfactant particles.

### Melting and Re-freezing: Temperature-driven Hysteresis

In this section we shall present an example of a hysteretic effect in a ternary mixture. Comparison of figures 7 and 8(a), show that the temperature has a strong influence on the properties of the steady state structure. Higher temperatures allow a broader range of microscopic configurations



(a)



(b)

Figure 8: System configuration from a temperature-driven hysteresis experiment after a)  $t = 6 \times 10^5$  and b)  $t = 10 \times 10^5$  time steps, for a ternary mixture of water, oil and  $M_2$  surfactant particles. An animation of this experiment can be found at: <http://tinyurl.com/lipid-CAs>

to be explored per unit time. However these structures are much less stable than those which emerge at low temperatures. At high temperatures, what we see at large scales is an average of a large number of possible configurations which are being adopted and then eradicated again in rapid succession. Alas the randomising effects of thermal energy reign. Since the absence of these effects allows the system to maintain its configuration over longer periods, we would expect that cooling a warm system should freeze in the approximate configuration which prevailed before the cooling began. So if we were to take a stationary cool system, heat it, allow it to relax and then cool it again, the final steady state will be different from that which results from leaving the system at a constant low temperature. It is this effect that gives glasses their amorphous structure. The relaxation time required for the molecular constituents of glasses to settle into their equilibrium positions is so long that they have the appearance of a liquid which has had its molecular motion suspended. We expect that we can create a similar effect

with our model system. We performed just such an experiment in which we initialised a simulation with identical parameters to those of the system shown in figure 7. It was allowed to relax for  $2 \times 10^5$  time steps before the temperature was linearly raised from  $T = 0.1 \rightarrow 0.4$  over a period of  $2 \times 10^5$  time steps. The system was then left for another  $2 \times 10^5$  time steps. The configuration at this point is shown in figure 8(a). The temperature was then returned to  $T = 0.1$  linearly over  $2 \times 10^5$  time steps and the system was allowed to relax once more. The final state of the system at the end of this process is shown in figure 8(b). The most prominent feature of figure 8 is that the high temperature state has indeed been ‘frozen in’ or quenched. However the alignment preferences of the surfactants have caused the monolayer to become much more rigid. Furthermore, because the total interface length has been reduced by the heating process, there are now more than enough surfactants to line it. As the temperature was lowered and potential minimisation became a stronger imperative, free drifting surfactants were forced out of the water regions and were adsorbed onto the monolayer. Some surfactants then started to form bilayer sections since joining the monolayer incurred a greater energy cost than extending a bilayer into the water region. This experiment showed that the geometric features of the configuration are not a simple function of state. They depend not only on the current conditions, but also on the system’s history. When the system is initialised at a low temperature and remains at that temperature, it retains remnants of its initial configuration. If the same system is heated and then cooled again, the final configuration reflects the state of the system at previous times when the environment was different. Not all details are retained but the differences between figures 7 and 8(b) highlight the fact that current environmental conditions alone are not sufficient to define the configuration of the system.

## Conclusions

We have presented a model of amphiphile structure formation which is both simple and shows qualitative agreement with experiment. As a foundation we adopted the framework of the artificial chemistry model of Ono and Ikegami (2001). By re-formulating the way that surfactants are represented in the model, we have given it the ability to successfully simulate some of the most common phases of amphiphilic systems. We have shown its ability to reproduce micelles, bilayers, reverse micelles and monolayers. Other phases including microemulsions have also been simulated.

Armed with the knowledge that protocells spontaneously formed in the original model of Ono and Ikegami (2001), and having established the basic lipid phenomenology of this new model version, our future work will involve simulating protocellular chemical systems in which the cells can adopt different membrane curvatures depending on the distribution of lipids in their membranes. This lipid distribution will directly impact their robustness and hence their ‘fitness’



with respect to other cells within the system. For example, possessing a small number of  $M_1$  and  $M_3$  particles would enable a cell to have a membrane composed of straight sections (primarily  $M_2$  particles) punctuated by high curvature corners ( $M_1$  particles on the inner side of the bilayer and  $M_3$  particles on the outer side). Such a membrane would have a significantly lower surface tension than one constructed purely from  $M_2$  particles. The ability to exchange resources and wastes involved in the synthesis of new lipid particles controls how efficiently a protocell can repair damage to its membrane and also how easily it can grow and divide to form a pair of daughter cells. Therefore this ‘full’ version of the model might give clues as to how mechanisms for cellular lipid homeostasis might emerge spontaneously. Further selection pressure could be placed upon the protocells by relaxing the assumption of a uniform, stationary environment. We should also be able to simulate self-reproducing micelles (Bachmann et al., 1992), and complex reaction-diffusion systems (Szymanski et al., 2011).

### Acknowledgements

This work was supported by an EPSRC Doctoral Training Centre grant (EP/G03690X/1).

### References

- Ayton, G. and Voth, G. A. (2002). Bridging microscopic and mesoscopic simulations of lipid bilayers. *Biophysical Journal*, 83(6):3357–3370.
- Bachmann, P. A., Luisi, P. L., and Lang, J. (1992). Autocatalytic self-replicating micelles as models for prebiotic structures. *Nature*, 357:57–59.
- Beard, J., Attard, G. S., and Cheetham, M. J. (2008). Integrative feedback and robustness in a lipid biosynthetic network. *Journal of the Royal Society Interface*, 5(22):533–543.
- Boghossian, B. M., Coveney, P. V., and Emerton, A. N. (1996). A lattice-gas model of microemulsions. *Proceedings of the Royal Society A*, 452(1948):1221–1250.
- Boghossian, B. M., Coveney, P. V., and Love, P. J. (2000). A three-dimensional lattice-gas model for amphiphilic fluid dynamics. *Proceedings of the Royal Society A*, 456(1998):1431–1454.
- Coveney, P. V., Emerton, A. N., and Boghossian, B. M. (1996). Simulation of self-reproducing micelles using a lattice-gas automaton. *Journal of the American Chemical Society*, 118:10719–10724.
- Frisch, U., Hasslacher, B., and Pomeau, Y. (1986). Lattice-gas automata for the navier-stokes equation. *Physical Review Letters*, 56(14):1505–1508.
- Kawakatsu, T., Kawasaki, K., Furusaka, M., Okabayashi, H., and Kanaya, T. (1994). Theories and computer simulations of self-assembling surfactant solutions. *Journal of Physics: Condensed Matter*, 6:6385–6408.
- Kier, L. B., Cheng, C.-K., and Testa, B. (1999). Cellular automata models of biochemical phenomena. *Future Generation Computer Systems*, 16(2-3):273 – 289.
- Larson, R. G., Scriven, L. E., and Davis, H. T. (1985). Monte carlo simulation of model amphiphile-oilwater systems. *The Journal of Chemical Physics*, 83(5):2411–2420.
- Lindblom, G., Wieslander, A., Sjoelund, M., Wikander, G., and Wieslander, A. (1986). Phase equilibria of membrane lipids for acholeplasma laidlawii: importance of a single lipid forming nonlamellar phases. *Biochemistry*, 25(23):7502–7510.
- Lyubartsev, A. (2005). Multiscale modeling of lipids and lipid bilayers. *European Biophysics Journal*, 35:53–61.
- Mayer, B., Köhler, G., and Rasmussen, S. (1997). Simulation and dynamics of entropy-driven, molecular self-assembly processes. *Physical Review E*, 55(4):4489–4499.
- Nilsson, M. and Rasmussen, S. (2003). Cellular automata for simulating molecular self-assembly. In Morvan, M. and Rémila, E., editors, *Discrete Models for Complex Systems*, volume AB of *DMTCS Proceedings*, pages 31–42. Discrete Mathematics and Theoretical Computer Science.
- Ono, N. (2005). Computational studies on conditions of the emergence of autopoietic protocells. *Biosystems*, 81(3):223–233.
- Ono, N. and Ikegami, T. (2001). Artificial chemistry: Computational studies on the emergence of self-reproducing units. In Kelemen, J. and Sosík, P., editors, *Proceedings of the 6th European Conference on Advances in Artificial Life*, pages 186–195. Springer-Verlag.
- Rothman, D. H. and Zaleski, S. (2004). *Lattice-Gas Cellular Automata: Simple Models of Complex Hydrodynamics*. Cambridge University Press.
- Szymanski, J., Gorecka, J. N., Igarashi, Y., Gizynski, K., Gorecki, J., Zauner, K.-P., and Planque, M. D. (2011). Droplets with information processing ability. *International Journal of Unconventional Computing*.
- Tresset, G. (2009). The multiple faces of self-assembled lipidic systems. *PMC Biophysics*, 2(3).
- Widom, B. (1986). Lattice model of microemulsions. *Journal of Chemical Physics*, 84(12):6943–6954.

# Evidence in the patent record for the evolution of technology using citation and PageRank statistics

M. A. Bedau,<sup>1,2,3,\*</sup> A. J. Buchanan,<sup>1</sup> D. W. Chalmers,<sup>1,4</sup> C. C. Francis,<sup>1</sup> N. H. Packard,<sup>2</sup> and N. M. Pepper<sup>4,1</sup>

<sup>1</sup>Reed College, Center for Advanced Computation, 3203 SE Woodstock Blvd., Portland OR 97202

<sup>2</sup>ProtoLife, Inc., 57 Post Street # 513, San Francisco CA 94104

<sup>3</sup>European School of Molecular Medicine, Campus IFOM-IEO, Via Adamello, 16 - 20139 Milan, Italy

<sup>4</sup>Lucky Sort, Inc., 8083 SE 13th Street #2, Portland OR 97202

\*Contact author: mab@reed.edu

## Abstract

We examine whether the process of technological innovation is an evolutionary process, in the sense that information that determines entities in the past is transmitted to entities in the future. We compare citation and PageRank statistics applied to data from the US patent record with data produced by certain non-evolutionary processes, captured by three classes of models that are driven respectively by what we term *random*, *preferential*, and *a priori* attachment. We make qualitative and quantitative comparisons of the cumulative citation curves produced by the patents and the three models, and find that random, *a priori*, and preferential attachment processes fail to explain certain significant patterns observed in the patent record—a result that corroborates the hypothesis that technological innovation is an evolutionary process.

## Is technological innovation evolutionary?

The advent of massive-scale systematic mining of aggregated social data, e.g., Michel et al. (2011), is transforming study of the evolution of science and technology. Earlier empirical study of the diffusion of innovations through social and economic markets (Rogers, 2003) can now be wedded with narrative theories the evolution of technology, e.g., Arthur (2009), and the analysis of innovation networks revealed in patent citation data (Jaffe and Trajtenberg, 2002). Interest in this topic rose sharply with the discovery that the growth and evolution of many kind of networks, including those consisting of citations among scientific papers (Redner, 1998; Barabasi and Albert, 1999; Lehmann et al., 2005) and patents (Valverde et al., 2007), exhibit power law behavior that can be modeled by various preferential attachment models. Here we take another look at this issue, and ask whether the characteristic dynamics of patent citations is well explained by three natural classes of models of the growth of patent citation networks.

There are a variety of reasons why those in artificial life might be interested in the evolution of technology. The process is driven by innovation, and understanding the role of innovation is essential to understand evolution, both in artificial life and in biology. If the technosphere—the set of all technological artifacts—displays a nontrivial form of evolu-

tion, it might itself be considered a sort living system, or a form of living technology (Bedau et al., 2010b,a).

The term “evolution” is usually used to describe the change of biological organisms over very long time periods, through a process that includes genetic variation of organisms from one generation to the next and natural selection based on survival of the fittest. This narrow biological view of evolution has been broadened to include long term changes in other non-biological systems; in recent literature one may find references to evolution of computer algorithms, evolutionary psychology, evolutionary history, cultural evolution, social evolution, sociocultural evolution, and technological evolution. But one may question the use of the term in all these contexts. A system may change over the long run, but when is that change properly termed evolution? Is there a well defined and empirically discernible difference between evolutionary change and non-evolutionary change?

Our view is that there is indeed a difference between evolutionary change and non-evolutionary change. For a system to evolve, it must be comprised of a population of entities, with a process for continual creation of new entities (the entities are analogs of biological organisms). The entities must be determined, at least in part, by some set of information (analog of an organism’s genome). And finally, there must be some process of selection taking place so that different entities are present to greater or lesser degree. We hold that such a system undergoes *evolutionary change* if and only if some of the information used in determining past entities persists and affects the determination of present and future entities (a form of heritability). If the present state of the population is substantially causally disconnected from all the determining information of previous populations, we would say that change is non-evolutionary. Biological evolution meets our definition of evolutionary change because the genetic information specifying present entities is copied from the genetic information from previous entities, perhaps modified by certain kinds of random mutations. Note that evolutionary change in the sense defined here covers both random genetic drift and darwinian evolution by natural se-

lection.

If the information determining a system's entities is measurable, the hypothesis that the system is evolving is testable through statistical analysis of those measurements. We consider the issue for technology, using citations in the patent record to examine whether or not technological change is evolutionary. In particular, we will consider the set of all technology that is represented by a patent granted by the United States Patent and Trademark Office (USPTO) during the period 1976-2010. Following our earlier work in this vein (Skusa and Bedau, 2002; Buchanan et al., 2011; Chalmers et al., 2010), we consider the set of patents as a changing population of entities, and we consider a patent to be "selected" whenever it is cited by another patent. When a new patent is issued, its immediate "ancestors" are considered to be the earlier patents that it cites, so citations are treated as the informational token indicating heritability. The most heavily cited patents are the key drivers of technological innovation.

Our previous work used various statistical measures based on citations to examine whether technological change is evolutionary. The citation-based statistics highlight various *a posteriori* narratives about superstar patents, *i.e.*, those that have especially high citation-based scores. Here we extend this analysis and examine other statistical indicators of a patent's persistence; specifically, we study second-order citations and PageRank, which can be viewed as depending on arbitrarily high-order past citations, and we examine whether these different statistical metrics alter the narratives of superstar patents. Finally, we measure the distance from the citation distributions observed in the patent record, to the characteristic citation distributions produced by three candidate generative models—a flat random attachment model, a preferential attachment model, and an *a priori* attachment model—and thus test the degree to which structure observed in the patent record is explained by our models. If the statistical character of data produced by a model is indistinguishable from the statistical character of the actual data, the actual data is well-explained by that model. We conclude that none of the three models considered effectively captures the structure found in the patent record.

## Statistics for quantifying evolution

In describing the citation dynamics of the patent record, we count citation events in which one patent cites another. A citation is a tuple: if  $p_1$  cites  $p_2$ , this first-order citation is the tuple  $(p_1, p_2)$ , and we say the relation  $c(p_1, p_2, t)$  holds, for  $p_1$  citing  $p_2$  at time  $t$ . (We sometimes assume an implicit existential quantifier over times, and simply speak of atemporal citation links between patents.) Previous work has examined the content of the citers of a patent  $p$  (Chalmers et al., 2010); in the present work, we focus on citation event tuples.

Let  $\overleftarrow{C}_1(p)$  be the set of first-order incoming citations of  $p$ :

$$\overleftarrow{C}_1(p) = \{(p', p) : \exists t, c(p', p, t)\}.$$

We can identify superstar patents by ranking all patents  $p$  according to their number of incoming citations,  $|\overleftarrow{C}_1(p)|$ . We can partition the set of incoming and outgoing citations into those received at some specific time  $t$ :

$$\overleftarrow{C}_1^t(p) = \{(p', p) : c(p', p, t)\}.$$

Generalizing, we let the *second-order citations* of a patent,  $\overleftarrow{C}_2(p)$ , be all the citation triples that end in  $p$ :

$$\overleftarrow{C}_2(p) = \{(p'', p', p) : \exists t, t', c(p'', p', t) \wedge c(p', p, t')\},$$

and partition a patent's second-order citations into those received at a specific time,  $t$ :

$$\overleftarrow{C}_2^t(p) = \{(p'', p', p) : \exists t' < t, c(p'', p', t') \wedge c(p', p, t)\}.$$

We regard a patent's *impact* on subsequent technological innovation as the cumulative weight of each citation event. For a counting function for an  $n^{\text{th}}$ -order citation,  $f^t(p_n, p_{n-1}, \dots, p_1, p)$ , we define a patent's  $n^{\text{th}}$ -order *impact*  $C_n^t(p)$  as the cumulative weight of citations to patent  $p$  up to time  $t$ :

$$C_n^t(p) = \sum_{t'=0}^{t'=t} \left( \sum_{(p_n, \dots, p_1) \in \overleftarrow{C}_n^{t'}(p)} f^t(p_n, \dots, p_1, p) \right). \quad (1)$$

The simplest version of a counting function is  $f^t(p_n, \dots, p_1, p) \equiv 1$ , in which case each citation in  $\overleftarrow{C}_n^t(p)$  is counted with equal weight.  $C_1^t(p)$  with this counting function is shown in the top of Figure 1. Buchanan et al. (2011) have shown that even this simple counting function reveals the main trends that remain prominent in the data after biases are removed.

We also iteratively calculate a patent's *PageRank*, which reflects the expected time that someone randomly surfing the patent citation network would spend visiting any given patent, as follows:

$$R_p^t(n) = (1 - d) + d \sum_{p' \in \overleftarrow{C}_1^t(p)} \left( \frac{R_{p'}^t(n-1)}{|\overleftarrow{C}_1^t(p)|} \right), \quad (2)$$

where  $n$  is the current iteration,  $d$  is a damping factor, and  $\overleftarrow{C}_1^t(p) = \{p' : c(p', p, t)\}$  and  $\overleftarrow{C}_1^t(p) = \{p' : c(p, p', t)\}$  are, respectively, the citers and citees of  $p$  up to time  $t$ . For all  $p$ , we let  $R_p^t(0) = 1$  and set  $d$  to 0.85, and perform 50 iterations—all as per convention in Page et al. (1999). A patent's PageRank changes over time, as the patent citation networks grows.

## Comparison of citations, second-order citations, and PageRank narratives

Earlier work operationalized high impact inventions as the most highly cited patents—termed patent “superstars.” Understanding superstar patents can help us understand what drives the evolution of technology in general. It turns out that superstar patents in the past few decades often involve PCR (the polymerase chain reaction that revolutionized contemporary biotechnology), inkjet printing, and stents (wire-mesh tubes that allow blocked coronary arteries to be repaired without open-heart surgery); see Buchanan et al. (2011) for details. The same methods also provide evidence that semiconductors, e-commerce, and wireless communication, for example, are also among the significant drivers of innovation during the last few decades (Chalmers et al., 2010). But the superstar status of PCR, ink-jet printing and stents remains a dominant pattern, so we illustrate our argument here by discussing those three key innovations.

Earlier work (Skusa and Bedau, 2002; Buchanan et al., 2011; Chalmers et al., 2010) compared citation counts of patents. Here, we extend the analysis to include two other natural statistics. We see in Figure 1 that the ten most cited patents appear among the top 100 when patents are ranked by second-order citations and PageRank, because more or less the same colored patents occur in all three plots. This shows that the PCR (red), inkjet printing (blue) and stents (green) narratives remain dominant when the patent record is analyzed by either citations, second-order citations, or PageRank. This rough correspondence between the three statistics tends to confirm that PCR, inkjet printing, and stents deservedly rank among the major technological innovations of the last thirty five years.

Nevertheless, the three different statistics do highlight different aspects of the patent record. For example, second-order citations correspond to the number of branches two levels down in the patent’s phylogenetic tree. The amount of green in the second-order citation plot shows the bushiness of the phylogenetic tree of the invention of stents—a conclusions confirmed by comparison of the phylogenetic trees of PCR, inkjet printing, and stents (data not shown). On the other hand, PageRank weights citations by the citer’s PageRank, so phylogenetic bushiness is insufficient by itself to boost PageRank. We see in the bottom of Figure 1 that stents (green) are significantly downplayed by PageRank, compared to PCR (red) and inkjet printing (blue). The observation that 25% of the top 20 patents ranked by second-order citation are about stents (green), while the top 100 patents include very few of the stent patents, might be connected with the earlier conclusion (Buchanan et al., 2011) that the stent patents are less “door-opening” than PCR and inkjet printing.

These qualitative conclusions are confirmed quantitatively by measuring the rank correlation between the patents when ranked by the three different statistics. The bottom

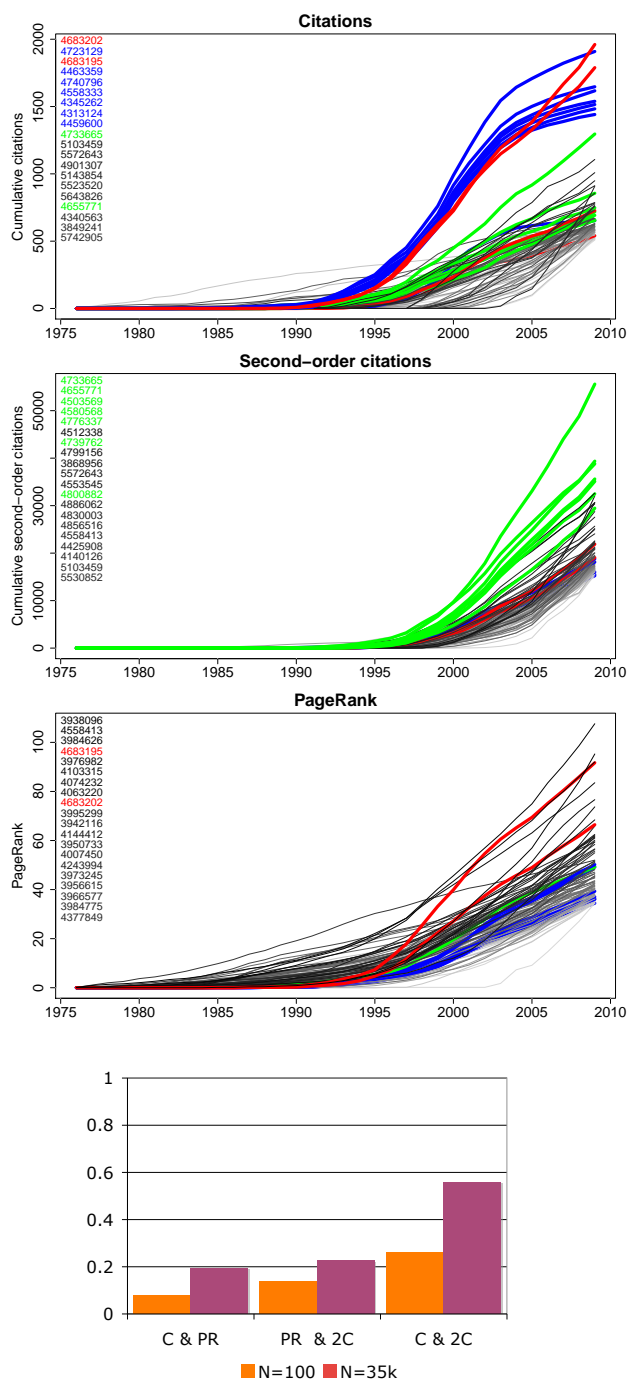


Figure 1: Top three panels: Dynamics in the patent record of cumulation first-order citations, cumulative second-order citations, and PageRank. Only the 100 highest-ranked patents are shown. Patents are color coded as follows: inkjet printing (blue), PCR (red), stents (green), other (gray). At the left are listed the patent numbers for the top twenty patents shown in each figure. Bottom: Rank correlations between the patents when ranked by citations and PageRank (C & PR), by PageRank and second-order citations (PR & 2C), and by citations and second-order citations (C & 2C), for the top 100 (orange) and the top 35k (purple) patents.

of Figure 1 reveals a significant difference between how the patents are ranked by the three statistics.

This conclusion can be generalized after examination of Figures 2 and 3. We can see in Figure 2 that most of the most heavily cited patents are also ranked highly by second-order citations and PageRank, because there is a lot of purple and brown at the top of the graphs. So, citations are highlighting something that is also picked up to some extent by second-order citations and PageRank. Similarly, we can see in Figure 3 that most of the patents with the highest PageRank are *not* also ranked highly by citations or second-order citations, because there is relatively little purple and brown at the top of the graphs. So, PageRank is highlighting something somewhat different than citations and second-order citations.

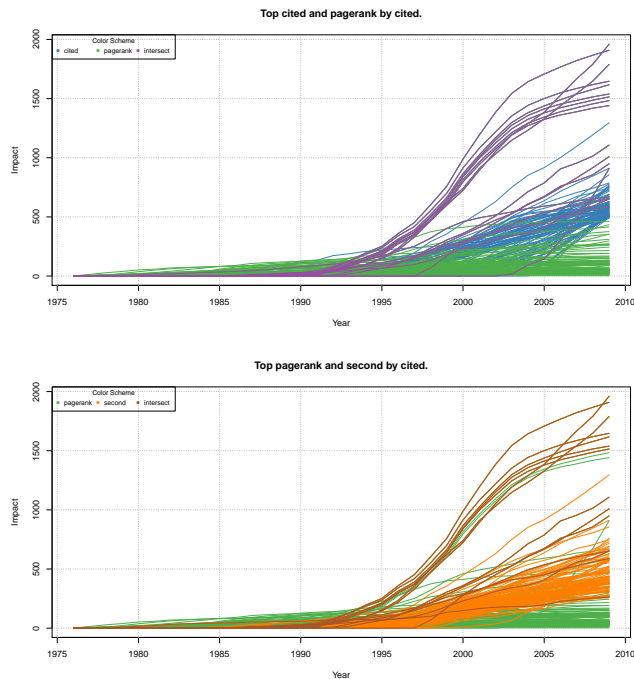


Figure 2: Above: First-order impact (cumulative citation) curves for the top 100 patents when ranked by citations or PageRank. Below: The same for the top 100 patents when ranked by PageRank or second-order citations. Patents are colored as follows: patents ranked among the top 100 only by citations (blue), those ranked among the top 100 only by PageRank (green), those ranked among the top 100 only by second-order citation (orange), those ranked among the top 100 by both citations and PageRank (purple), those ranked among the top by both PageRank and second-order citation (brown).

## Comparison with three models

We test whether a system is producing evolutionary change by comparing its citation network with the citation networks produced by various hypothetical non-evolutionary model systems, consisting of processes that generate citation networks, with different degrees of structure built into the processes. We test the likelihood of the hypotheses that the actual patent citation network was produced by a process embodied by those model systems, by comparing the statistical character of the citation networks produced by the model systems with that of the actual data. If, for a particular model system, the statistical character of the citation network were indistinguishable, we would say that the actual data is well-modeled by that model system.

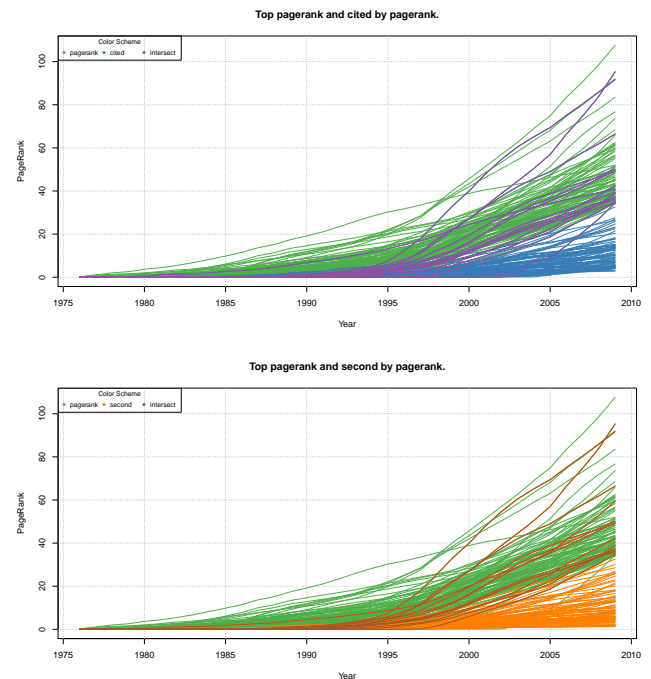


Figure 3: Above: First-order impact (cumulative citation) curves for the top 100 patents when ranked by citations or PageRank. Below: The same for the top 100 patents when ranked by citations or PageRank. Patents are colored as in Figure 2.

All the models work in the same basic fashion. Rather than the actual references made, each patent's citations are semi-randomly assigned, creating a new network of patent citations. The models differ in how citations are assigned, from completely randomly to significantly favoring certain patents as described below. All of these models are *non-evolutionary* because citation do not transfer information from one patent to another; information about a past patent is not inherited when a later patent cites it. Differing from the actual data, there is no lag between application and issue

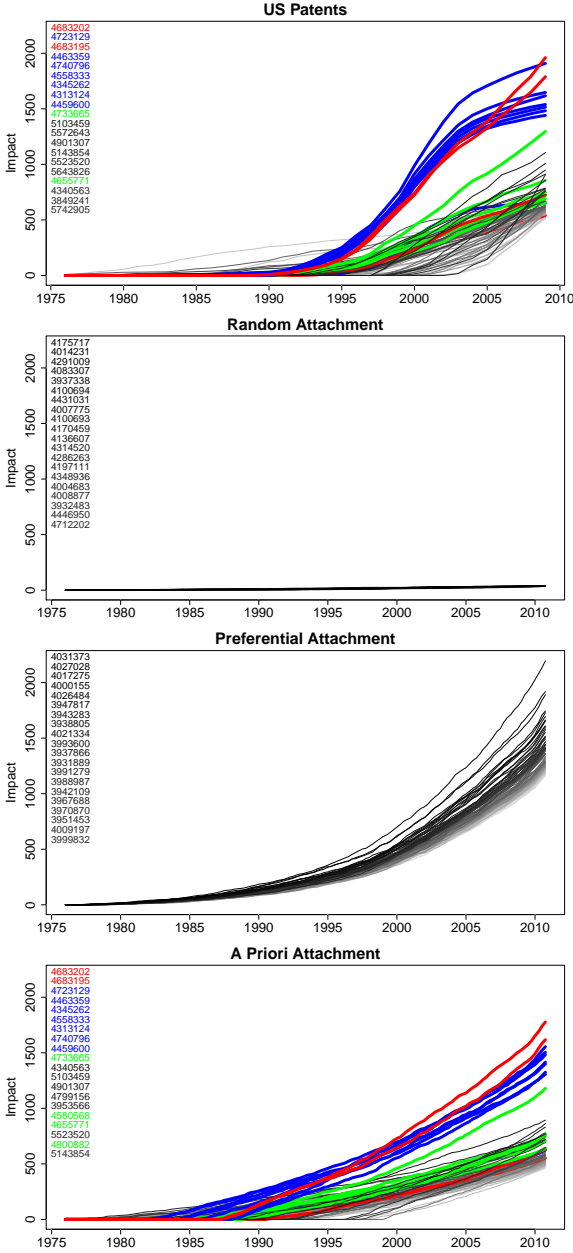


Figure 4: Curves of first-order impact (cumulative citations) for the 100 most-cited patents produced by four different processes: Top: patents, including those about inkjet printing (blue), PCR (red), and stents (green). Upper Middle: flat neutral model. Lower Middle: preferential attachment model,  $PW = 2.0$ . Below: *a priori* attachment model (colors mirror US patents),  $APW = 0.9$ .

date in the models. An actual patent faces a delay of anywhere from two to four years on average between application and issuance, when other patents may begin to cite it. In the models, patents become eligible for citation immediately after their own citations have been reassigned. The models also differ from the actual data in allowing a patent to cite the same patent more than once. This eases implementation of models, and is not expected to significantly change their behavior.

### Random attachment model

In the random attachment model, the chance that a given patent has of being cited at a given time is equal to the chance that any other patent has of being cited at that time (Skusa and Bedau, 2002; Buchanan et al., 2011; Chalmers et al., 2010). Of course, the set of patents that available to be cited continually grows over time, and the number of incoming citations can fluctuate over time. More precisely, given a citing patent  $c$ , the chance of an earlier patent  $e$  receiving a citation is  $1/N$ , where  $N$  is the number of patents in our dataset issued before patent  $c$ . The random attachment model prevents the information determining past patents from having any effect on present or future patents that cite them.

### Preferential attachment model

We define a family of preferential attachment models,  $PA-k$ , where  $k$  is a parameter equal to the weight of preference parameter in the model. Here, we set  $k = 2.0$ , which results in the most frequently cited patents receiving approximately the same number of citations as most frequently cited real patents. More precisely, each patent,  $p'$ , is assigned a weight,  $w$ , which affects the chance of later patents citing it. That weight is a linear function of the number of citations received so far:

$$w' = b + kr \quad (3)$$

where  $b$  is the weight of a patent with no citations and  $r$  is the number of citations received to that point. Here,  $b$  was set to 1.0. A patent's chance of being cited is

$$\frac{w'_p}{\sum_{p \in M} w_p}$$

where  $M$  is the set of patents in our dataset issued prior to  $c$ . In the preferential attachment model, there is information about a patent that directly affects its probability of being cited—specifically, the number of citations it has already received. But that information is not inherited by the other patents that cite it.

### *A priori* attachment model

We define a family of *a priori* attachment models,  $AP-k$ , where  $k$  is a parameter equal to the weight of *a priori* attachments. These models choose which patents to cite by



sampling from a fixed “*a priori*” distribution of the relative “value” of each patent. Here, we choose this distribution to be given by the actual number of citations a patent has received by the end of 2010. We choose  $k = 0.9$ , because this leads to similar maximum numbers of citations received in the model and the actual data. Each patent’s weight for deciding citation assignments is given by

$$w' = a^k \quad (4)$$

where  $a$  is the patent’s *a priori* weight, in this case the number of citations received by 2011. Each patent’s chance of being cited is given by the ratio of its weight to the total weight, as in the preferential attachment model. As in the previous two models, in the preferential attachment model information used to determine a patent is not inherited by later patents that cite it.

### Comparison of scaled citation curves

We investigate whether the curves of cumulative citation counts of patents have a distribution of shapes that is significantly different from the distribution of shapes of cumulative citation curves produced by random attachment, preferential attachment, and *a priori* attachment.

The impact curves produced by random attachment, preferential attachment, and *a priori* attachment have distinctive characteristic shapes. The models produce curves with different characteristic shapes. Furthermore, none of the models produce curves very much like those displayed by patents. The different shapes are readily apparent in Figures 4 and 5.

Figure 4 compares the typical impact (cumulative citation) curves in the patent record with curves produced by three different null hypotheses. The 10 most cited patents fall into inventions: PCR (red), inkjet printing (blue), and stents (green). Note that the random attachment process produces curves of which the highest are over an order of magnitude smaller than the highest actual patent curves, even though both processes produce the same total number of citations. By contrast, we note that preferential and *a priori* attachment produce curves of about the same size as the patents; however, the similar sizes is a direct consequence of how we set the weights in those two modes. (Current work includes estimating the weights from the patent record itself.) Note that none of the colored patents appear in the random and preferential attachment curves; this is because we plot only the top 100 most cited patents, and the most cited patents in the random and preferential attachment processes are chosen from a uniform distribution, so the probability of a colored patent being in the top 100 is very low (about  $3 \times 10^{-6}$ ). By contrast, since we set a patent’s *a priori* probability of being cited by its actual citation count in 2010, the patents that are actually most cited—the colored patents—are expected to be the most cited patents produced by the *a priori* attachment model.

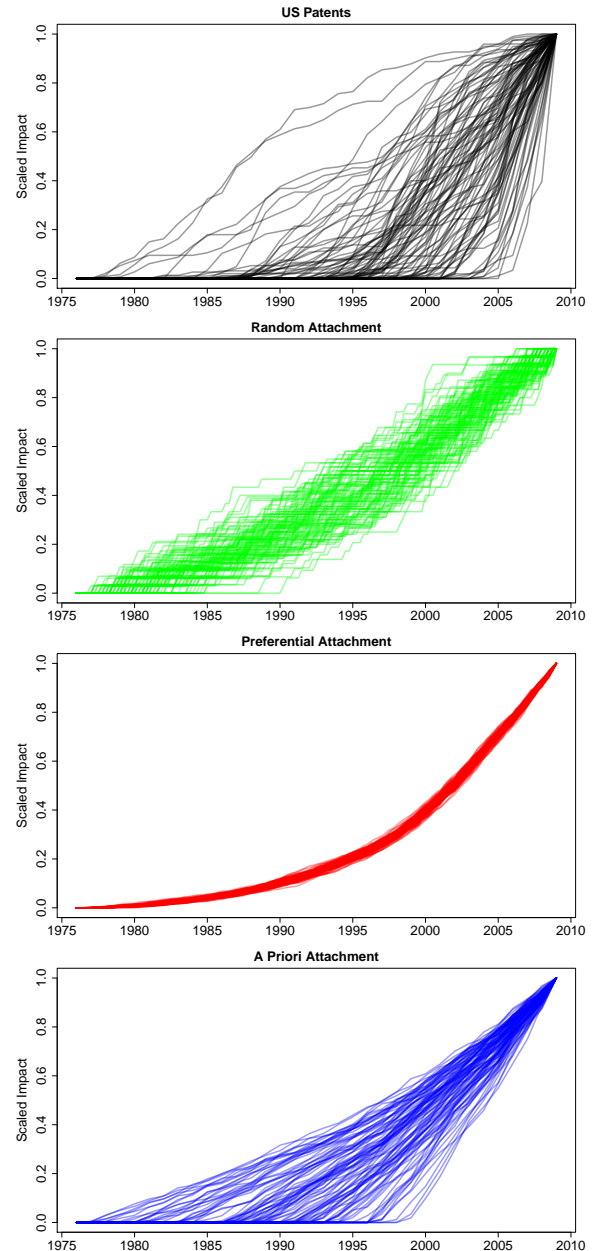


Figure 5: Curves of first-order impact (cumulative citations), scaled to the interval  $[0, 1]$ , for the 100 most cited patents (top, black), the 100 most cited patents produced by the random attachment model (green), preferential attachment (red) and *a priori* attachment (bottom, blue).

A good way to visualize the characteristic shape of the impact curves produced by the different processes is to mask their size differences by scaling them to the interval  $[0, 1]$ . Figure 5 shows that the shape of the curves of patents (black, top panel) is qualitatively different from the shape of the curves produced by random attachment (green, second panel), by preferential attachment (red, third panel), and by *a priori* attachment (blue, bottom panel). The random attachment curves (green) all increase at roughly the same expected linear rate. The preferential attachment curves (red) all have the same expected shape, which increases in lock-step with the number of patents being issued (Buchanan et al., 2011). The *a priori* attachment curves all increase linearly, starting immediately once a patent is issued. By comparison, the actual patents (black) display a wide variety of shapes, many of which are not found in the curves produced by the three models. This is strong evidence that the process producing the actual patent citation networks is not random attachment, preferential attachment, or *a priori* attachment.

The difference between these families of curves can be quantified using a measure of statistical distance between distributions, where each family corresponds to samples from the distribution describing that family. Statistical distance between univariate distributions is well measured by the Kolmogorov-Smirnov distance measure, which is simply the maximum difference between the cumulative distribution functions of the two distributions. The citation and Page rank curves, however, are not univariate. They are typically sampled each quarter of the year, which results in 132 measurements for each patent; *i.e.*, each curve is represented by a point in a 132-dimensional space, where each dimension represents the value of a statistic for one of the 132 quarters.

For distributions in higher dimensions, *i.e.*, multivariate distributions, distance measures are not quite so straightforward as in the univariate case. This is because in  $d$ -dimensional spaces the cumulative distribution depends on a choice of ordering of the coordinates, and there are  $2^d - 1$  possible orderings. One can define a distance as the supremum over all orderings, which is cumbersome to compute, or estimate based on a sample of orderings, or choose a particular ordering. We have used an `medcf` R package that computes a multivariate empirical cumulative distribution function for computation of a Kolmogorov-Smirnov distance.

Before constructing the cumulative distribution function, however, it is useful to reduce the dimension of the curves by fitting them to orthogonal polynomials. We used the first four Legendre polynomials, reducing each 132-value curve to a point in a five dimensional space. We then used `medcf` to construct the empirical cumulative distribution functions in the five dimensional space for each family of curves, and sampled that cdf eight values in each dimen-

sion ( $8^5 = 32,786$  samples), equally spaced over the range obtained by taking the minimum and maximum values for each coordinate. The estimated distance between two distributions is then the maximum value of the difference between the cdf samples for the two distributions, over all the sampled points. For example, the estimated Kolmogorov-Smirnov distance between two families of points in five dimensions, one produced by a Gaussian with mean zero and standard deviation one, and the other produces by a Gaussian with the same standard deviation, but a displaced mean, is about 0.4 if the two Gaussians are separated by two standard deviations.

Figure 6 shows the KS distances from the distribution of scaled cumulative citation curves in the patents for the curves produced by the *a priori* attachment model (blue), the preferential attachment model (red), and the random attachment model (green). The KS distance between the patents and both preferential and random attachment models quantitatively confirms what the eye can see in Figure 5: The shape of the curves is significantly different. At the same time, the KS distance is much less for the curves produced by *a priori* attachment, which also confirms what the eye tends to see in Figure 5.

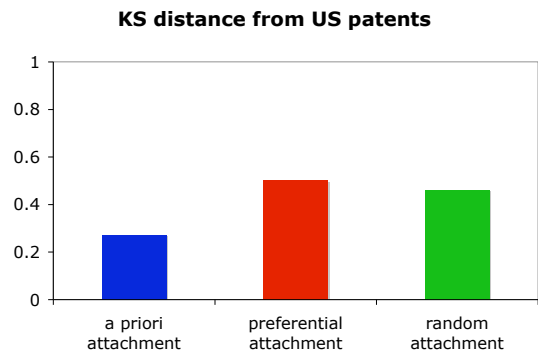


Figure 6: KS distances between distributions of cumulative citation curves scaled to  $[0, 1]$ . Curves produced by the random, *a priori*, and preferential attachment models are compared with the curves derived from the patent record.

## Conclusions

The statistical exploration of the patent record supports two main conclusions about the process of technological innovation. First, the superstars of citations show up among the superstars of second-order citations, and also but to a lesser extent among the superstars of PageRank. The different statistics highlight somewhat different aspects of the patent record, but roughly the same stories come through with all



the statistics. In particular, PCR, inkjet printing, and stents are among the dominant innovations from all three perspectives. This implies that the superstar status of PCR, inkjet printing, and stents is not just a quirk of citation statistics; instead, those innovations are genuinely among the robust and dominant drivers of innovation in the past forty years.

Second, the shape of cumulative citation curves produced by the patents is significantly different from the shapes of the curves produced by random, *a priori*, and preferential attachment processes. This qualitative difference is corroborated by the significant KS distance between those same sets of curves. This implies that the process producing technological innovation is a fundamentally different from the processes produced by random, *a priori* and preferential attachment. This conclusion tends to be corroborated by earlier work that analyzed the size (rather than shape) of cumulative citation curves (Skusa and Bedau, 2002; Buchanan et al., 2011). Current work includes examining whether this conclusion is affected if the *a priori* and preferential attachment models are driven with an empirically estimated distribution of the probability of receiving a citation as a function of both the number of citations already accumulated and the age of the patent, in line with augmenting preferential attachment with “death” (Lehmann et al., 2005) or “aging” (Valverde et al., 2007).

The difference between the citation statistics found in the patent record and those produced by random, *a priori*, and preferential attachment processes suggests a further conclusion. Random, *a priori*, and preferential attachment are all non-evolutionary processes, because the information that determines the entities in the population is not inherited when those entities are cited in the future. By contrast, intuitively it seems that citations between patents *do* represent the propagation into the future of information about past technology, which implies that technological change is an evolutionary process. The difference between the statistical features of the data produced by the three classes of models and by the data found in the patent record corroborates the hypothesis that technological innovation is a genuinely evolutionary process.

## References

- Arthur, W. B. (2009). *The Nature of Technology: What it is and How it Evolves*. Simon & Schuster, Free Press.
- Barabasi, A. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512.
- Bedau, M. A., Hansen, P. G., Parke, E., and Rasmussen, S., editors (2010a). *Living Technology: 5 Questions*. Automatic Press/VIP, Copenhagen.
- Bedau, M. A., McCaskill, J. S., Packard, N. H., and Rasmussen, S. (2010b). Living technology: Exploiting life’s principles in technology. *Artificial Life*, 16(1):89–97. PMID: 19857142.
- Buchanan, A., Packard, N. H., and Bedau, M. A. (2011). Measuring the evolution of the drivers of technological innovation in the patent record. *Artificial Life*, 17:109–122.
- Chalmers, D., Francis, C. C., Pepper, N., and Bedau, M. A. (2010). High-content words in patent records reflect key innovations in the evolution of technology. In Fellermann, H., Drr, M., Hanczyc, M. M., Laursen, L. L., Maurer, S., Merkle, D., Monnard, P.-A., Stoy, K., and Rasmussen, S., editors, *Proceedings of Artificial Life XII*, pages 838–845. MIT Press.
- Jaffe, A. B. and Trajtenberg, M. (2002). *Patents, Citations, and Innovations: A Window on the Knowledge Economy*. MIT Press, Cambridge, Mass.
- Lehmann, S., Jackson, A. D., and Lautrup, B. (2005). Life, death and preferential attachment. *Europhysics Letters*, 69:298–303.
- Michel, J.-B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., Pickett, J. P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M. A., and Aiden, E. L. (2011). Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182.
- Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.
- Redner, S. (1998). How popular is your paper? an empirical study of the citation distribution. *The European Physical Journal B*, 4:131–134.
- Rogers, E. M. (2003). *Diffusion of Innovations*. Free Press, New York, 5th edition.
- Skusa, A. and Bedau, M. A. (2002). Towards a comparison of evolutionary creativity in biological and cultural evolution. In Standish, R., Bedau, M. A., and Abbass, H. A., editors, *Artificial Life VIII*, pages 233–242. MIT Press, Cambridge, MA.
- Valverde, S., Sole, R. V., Bedau, M. A., and Packard, N. (2007). Topology and evolution of technology innovation networks. *Physical Review E*, 76:056118.1 – 056118.7.

# Theory and Practice of Optimal Mutation Rate Control in Hamming Spaces of DNA Sequences

Roman V. Belavkin<sup>1</sup>, Alastair Channon<sup>2</sup>, Elizabeth Aston<sup>2</sup>, John Aston<sup>3</sup> and Christopher G. Knight<sup>4</sup>

<sup>1</sup>School of Engineering and Information Sciences, Middlesex University, London NW4 4BT, UK

<sup>2</sup>Research Institute for the Environment, Physical Sciences and Applied Mathematics, Keele University, ST5 5BG, UK

<sup>3</sup>Department of Statistics, University of Warwick, CV4 7AL, UK

<sup>4</sup>Faculty of Life Sciences, University of Manchester, M13 9PT, UK

r.belavkin@mdx.ac.uk, {a.d.channon,e.j.aston}@epsam.keele.ac.uk, j.a.d.aston@warwick.ac.uk, chris.knight@manchester.ac.uk

## Abstract

We investigate the problem of optimal control of mutation by asexual self-replicating organisms represented by points in a metric space. We introduce the notion of a relatively monotonic fitness landscape and consider a generalisation of Fisher's geometric model of adaptation for such spaces. Using a Hamming space as a prime example, we derive the probability of adaptation as a function of reproduction parameters (e.g. mutation size or rate). Optimal control rules for the parameters are derived explicitly for some relatively monotonic landscapes, and then a general information-based heuristic is introduced. We then evaluate our theoretical control functions against optimal mutation functions evolved from a random population of functions using a meta genetic algorithm. Our experimental results show a close match between theory and experiment. We demonstrate this result both in artificial fitness landscapes, defined by a Hamming distance, and a natural landscape, where fitness is defined by a DNA-protein affinity. We discuss how a control of mutation rate could occur and evolve in natural organisms. We also outline future directions of this work.

## Introduction

The problem of optimal mutation rate has been studied for a long time (e.g. see Eiben et al., 1999; Ochoa, 2002; Falco et al., 2002; Cervantes and Stephens, 2006; Vafaei et al., 2010, for reviews). It relates directly to optimisation of genetic algorithms (GAs) in operations research and engineering problems (i.e. meta-heuristics). It is also related to some fundamental questions in evolutionary theory about the role of mutation in adaptation and biological mechanisms of DNA repair and mutation control.

As noted by Eiben et al. (1999), there are two trends in optimisation of parameters in GAs — optimal parameter setting and optimal parameter control. In the former, one looks for an optimal value of a parameter, which is then kept constant. Thus, Mühlenbein (1992) proposed mutation rate  $\mu = 1/l$ , where  $l$  is the length of sequences. The value  $1/l$ , as was pointed out by Ochoa et al. (1999), is related to the error threshold (Eigen et al., 1988). However, while mutation rate  $1/l$  can give satisfactory performance in some problems, the advantages of using a variable rate were becoming obvious to many researchers, leading to the problem

of optimal parameter control. In particular, Ackley (1987) suggested that mutation probability is analogous to temperature in simulated annealing, and should decrease with time. A gradual reduction of mutation rate was also proposed by Fogarty (1989). In a pioneering work, Yanagiya (1993) used Markov chain analysis of GAs to show that in any problem there exists a sequence of optimal mutation rates maximising the probability of obtaining global solution at each generation. A significant contribution to the field was made by Bäck (1993), who suggested that mutation rate  $\mu$  should depend on fitness values rather than time. Recently, Vafaei et al. (2010) used numerical methods to optimise a mutation operator based on the Markov chain model of GA by Nix and Vose (1992). The complexity of this model, however, restricts the application of this method to small spaces and populations. Thus, the precise form of the optimal mutation rate control, as well as question about the existence of such a control in the general case, remain open problems. These problems are extremely important not only for applications of GAs, but also for biology and evolutionary theory.

In biological systems, mutation, unlike natural selection, is an evolutionary process controlled, to a degree, by the organism. This control is primarily seen in highly refined DNA repair and replication machinery (e.g. Hakem, 2008). This ensures both that physical damage to genetic material is repaired and that, in the process of cell division, the newly synthesised copies of DNA faithfully reproduce the parental sequence. The result is that biological mutation rates are very low: DNA-based organism values typically being 1/300 per genome per replication, which, for genomes frequently in the range between  $10^6$  and  $10^{10}$  base-pairs, means extremely faithful repair and replication (Drake et al., 1998). Nonetheless, this observation also implies that biological mutation rates per base-pair are not minimised, since widely varying genome sizes imply very different per-base-pair rates. At a mechanistic level, some organisms do exist, such as the bacterium *Deinococcus radiodurans* with substantially more developed DNA repair or replication mechanisms than closely related species (Cox et al., 2010), implying that mutation rates elsewhere at least are not minimised.

Genetic variation also exists in mutation rates within species and in the way mutation rate changes with environment for a single genotype (Bjedov et al., 2003). Therefore, mutation rates and their variation are potentially subject to biological evolution themselves. Thus, while mutation rates are only a part of the biological evolutionary process, they merit examination independent of the vicissitudes of selection that are imposed on their products, which is what we address.

Our approach is based on theories of optimal control and information. However, we believe that the key to finding solutions that are relevant not only for engineering, but also for biology, is understanding the relation between a representation space, which is a discrete space of genotypes, and its (pre)-ordering by phenotypic fitness. Biology typically understands this relation via landscape metaphors, used in a variety of ways (e.g. classically adaptive landscapes of Wright, 1932, and ‘epistatic’ landscapes of Waddington, 1957). However, while the underlying elements, particular alleles of genes, are acknowledged as discrete, these landscapes have almost uniformly been theorised (and visualised) in continuous space following Fisher (1930). This is problematic when one comes to the mechanistic basis of biological evolution in discrete DNA mutations. Attempts are being made to reconcile such continuous models with individual DNA mutations (Orr, 2005). However, hitherto, these attempts have maintained a continuous view of the landscape space, in contrast to the reality of its discrete domain. Discrete views have typically been restricted to abstracted biological systems, such as aptamer (Knight et al., 2009) or RNA structure evolution, where landscape analogies can be dropped in favour of networks of sequences (Cowperthwaite and Meyers, 2007) which do not lend themselves to consideration of variable mutation sizes.

This work presents elements of a theory on optimisation of asexual reproduction by a mutation rate control together with its experimental evaluation. We introduce the notion of relatively and weakly monotonic fitness landscapes, and then develop the necessary machinery for Hamming spaces of sequences with arbitrary alphabets, which are particularly relevant in biology. Then we evolve mutation rate control functions using a meta genetic algorithm, and show that they closely match our theoretical predictions.

## Theory

Let  $\Omega$  be a countable set of all possible individuals  $\omega$  and  $f : \Omega \rightarrow \mathbb{R}$  be a fitness function. Assuming that fitness value  $x = f(\omega)$  is the only information available, let  $P(x_{s+1} | x_s)$  be the conditional probability of an offspring having fitness value  $x_{s+1}$  given that its parent had value  $x_s$  at generation (time)  $s$ . This Markov probability can be represented by a left stochastic matrix  $T$ , and if  $P(x_{s+1} | x_s)$  does not depend on  $s$  (i.e.  $T$  is stationary), then  $T^t$  defines a linear transformation of distribution  $p_s := P(x_s)$  of fitness values at time  $s$  into distribution  $p_{s+t} := P(x_{s+t})$  of fitness values

after  $t \geq 0$  generations:

$$p_{s+1} = T p_s = \sum_{x_s} P(x_{s+1} | x_s) P(x_s) \Rightarrow p_{s+t} = T^t p_s$$

We denote the expected fitness at generation  $s$  as

$$\mathbb{E}\{x_s\} := \sum_{x_s} x_s P(x_s)$$

If  $\mathbb{E}\{x_{s+t}\} \geq \mathbb{E}\{x_s\}$ , then individuals have adapted.

Suppose that the transition probability  $P_\mu(x_{s+1} | x_s)$  depends on a control parameter  $\mu$ , so that the Markov operator  $T_{\mu(x)}$  depends on the control function  $\mu(x)$ . Then the expected fitness  $\mathbb{E}_{\mu(x)}\{x_{s+t}\}$  also depends on  $\mu(x)$ . We interpret  $\mu(x)$  as a control function that parents use in reproduction to maximise expected fitness of their offspring based on the value of their own fitness. A particular example we shall consider here is when  $\mu$  is the mutation rate parameter.

If  $\Omega$  is the space  $\mathcal{H}_\alpha^l := \{1, \dots, \alpha\}^l$  of sequences of length  $l$  and  $\alpha$  letters, then by mutation we understand here a process of independently changing each letter in a parent sequence to any of the other  $\alpha - 1$  letters with probability  $\mu/(\alpha - 1)$ . This is point mutation, the simplest form of mutation defined by one parameter  $\mu$ , called the *mutation rate*.

The main result that we present in this paper is a mutation rate control function, which is approximately optimal for maximising expected fitness  $\mathbb{E}\{x_{s+t}\}$  in landscapes  $f(\omega)$  that are locally monotonic relative to the Hamming metric (this property will be defined later). This mutation rate function corresponds to the cumulative distribution function (CDF)  $P_e(x_r > x)$ ,  $r \in [s, s+t]$ , computed from empirical distribution  $P_e(x_r)$  of observed fitness values  $x_r$  over the period  $[s, s+t]$ :

$$\mu_e(x) = P_e(x_r > x) = \sum_{x_r > x} P_e(x_r) \quad (1)$$

We refer to this function as *informed mutation rate*, because it uses information communicated by random variable  $x$ . We first present the theory and assumptions behind this heuristic. Then we evaluate it against nearly optimal mutation functions, evolved using a meta genetic algorithm both for artificial and natural fitness landscapes.

## Problem Definition

Formally, an optimal control function (e.g. an optimal mutation rate function) is  $\bar{\mu}(x)$  achieving the following optimal (supremum) value:

$$\bar{x}(\lambda) := \sup_{\mu(x)} \{\mathbb{E}_{\mu(x)}\{x_{s+t}\} : t \leq \lambda\} \quad (2)$$

Here,  $\lambda$  represents a time constraint. Function (2) is non-decreasing and has the following inverse

$$\bar{x}^{-1}(v) := \inf_{\mu(x)} \{t \geq 0 : \mathbb{E}_{\mu(x)}\{x_{s+t}\} \geq v\} \quad (3)$$

Here,  $v$  is a constraint on the expected fitness at  $s + t$ . Thus,  $\bar{x}(\lambda)$  is the maximum adaptation in no more than  $\lambda$  generations;  $\bar{x}^{-1}(v)$  is the minimum (infimum) number of generations required to achieve adaptation  $v$ .

Optimal solutions  $\bar{\mu}(x)$ , defined by function (2), depend on the constraint  $t \leq \lambda$ . We are interested in solutions for  $\lambda$  that is large enough to achieve the maximum expected fitness  $\mathbb{E}\{x_{s+t}\} = \sup f(\omega)$ . This can be represented dually by function (3) with constraint  $v = \sup f(\omega)$ . We note that  $\bar{x}(\lambda) = \sup f(\omega)$ , if  $\lambda = \infty$ . However, generally  $\bar{x}^{-1}(v) \leq \infty$ , even if  $v = \sup f(\omega)$ . Thus, our objective is to derive one optimal control function  $\bar{\mu}(x)$  that can be used by each individual parent based on their fitness value throughout the entire ‘evolution’  $[s, s + t]$ . We note also that our formulation uses only the values of fitness, and therefore it extends to the case where  $f(\omega)$  is time-variable.

Specific expressions for  $P_\mu(x_{s+1} | x_s)$ , defining  $T_{\mu(x)}$ , can be learnt or derived analytically from the domain  $\Omega$  and its structure. The operator  $T_{\mu(x)}$  contains all information required to compute optimal values (2) and (3). Thus, in principle, one can find an optimal control function  $\bar{\mu}(x)$ , if the family of operators  $T_{\mu(x)}$  is known. For example, considering values  $x \geq v$  as absorbing states, one can use  $T_{\mu(x)}$  to compute the fundamental matrix of the corresponding absorbing Markov chain and minimise the expected convergence time to the absorbing states. Solving the complete optimisation problem, however, can be an intractable task. We shall formulate additional assumptions that will allow us to solve the problem for some important cases.

## Relatively Monotonic Landscapes

First, we shall make some assumptions about fitness  $f(\omega)$ , which on one hand will generalise and clarify the terms ‘smooth’ and ‘rugged’ fitness landscape, and on the other hand will allow us to obtain expressions for  $P_\mu(x_{s+1} | x_s)$ . In particular, we assume that there exists optimal individual  $\top \in \Omega$  (not necessarily unique) such that  $\sup f(\omega) = f(\top)$ . This is always true if  $\Omega$  is finite. Also, we shall equip  $\Omega$  with a metric  $d : \Omega \times \Omega \rightarrow [0, \infty)$ , so that similarity between  $a$  and  $b \in \Omega$  can be measured by  $d(a, b)$ , and assume that there is a relation between the metric  $d$  and the fitness function  $f$ . In particular, we define  $f$  to be monotonic relative to  $d$ .

**Definition 1** (Monotonic landscape). Let  $(\Omega, d)$  be a metric space, and let  $f : \Omega \rightarrow \mathbb{R}$  be a function with  $f(\top) = \sup f(\omega)$  for some  $\top \in \Omega$ . We say that  $f$  is *locally monotonic* (locally isomorphic) relative to metric  $d$  if for each  $\top$  there exists a ball  $B(\top, r) := \{\omega : d(\top, \omega) \leq r\} \neq \{\top\}$  such that for all  $a, b \in B(\top, r)$ :

$$-d(\top, a) \leq -d(\top, b) \implies (\iff) f(a) \leq f(b)$$

We say that  $f$  is *monotonic* (isomorphic) relative to  $d$  if  $B(\top, r) \equiv \Omega$ .

**Example 1** (Needle in a haystack). Let  $f(\omega)$  be defined as

$$f(\omega) = \begin{cases} 1 & \text{if } d(\top, \omega) = 0 \\ 0 & \text{otherwise} \end{cases}$$

This fitness landscape is often used in studies of GA performance. A two-valued landscape is used to derive error threshold and critical mutation rate, and elements  $\top$  are referred to as the *wild type*. Such  $f$  is locally monotonic relative to any metric, if for each  $\top \in \Omega$  there exists  $B(\top, r) \neq \{\top\}$  containing only one  $\top$ . Then conditions of the definition above are satisfied in all such  $B(\top, r) \subset \Omega$ . If  $\Omega$  has unique  $\top$ , then the conditions are satisfied for  $B(\top, \infty) = \Omega$ . In a two-valued landscape, optimal function  $\bar{\mu}(x)$  for any  $\lambda$  in (2) is defined by maximising the one-step transition probability  $P_\mu(x_{s+1} = 1 | x_s)$ .

**Example 2** (Negative distance to optimum). If  $f$  is isomorphic to  $d$ , then one can replace fitness  $f(\omega)$  by the negative distance  $-d(\top, \omega)$ . The number of values of such  $f$  is equal to the number of spheres  $S(\top, r) := \{\omega : d(\top, \omega) = r\}$ . One can easily show also that when  $f$  is isomorphic to  $d$ , then there is only one  $\top$  element:  $f(\top_1) = f(\top_2) \iff d(\top_2, \top_1) = d(\top_2, \top_2) = 0 \iff \top_1 = \top_2$ .

In monotonic landscapes, spheres  $S(\top, r)$  cannot contain individuals with different fitness. We can generalise this property by *weak* or  $\varepsilon$ -monotonicity, which requires that the variance of fitness within individuals of each sphere  $S(\top, r)$  is small or does not exceed some  $\varepsilon \geq 0$ . These assumptions allow us to replace fitness  $f(\omega)$  by negative distance  $-d(\top, \omega)$ , and derive expressions for transition probability  $P_\mu(x_{s+1} | x_s)$  using topological properties of  $(\Omega, d)$ .

Monotonicity of  $f$  depends on the choice of metric, and one can define different metrics on  $\Omega$ . Fitness landscapes that are at least weakly locally monotonic relative to the Hamming metric seem biologically plausible given the abundance of neutral mutations in nature and redundancy in the translation of DNA to protein sequences. Thus, we focus our attention on the case when  $\Omega$  is a Hamming space.

## Mutation and Adaptation in a Hamming Space

First, we outline a model of asexual reproduction in metric space  $(\Omega, d)$ , and define the relation of parameter  $\mu$  to topology on  $\Omega$ . This model is a generalisation of Fisher’s geometric model of adaptation in Euclidean space (Fisher, 1930). Then we shall specialise this to a Hamming space.

Let individual  $a$  be a parent of  $b$ , and let  $d(a, b) = r$ . We consider single-parent reproduction as a transition from parent  $a$  to a random point  $b$  on a sphere:  $b \in S(a, r)$ . We refer to  $r$  as a *radius of mutation*. Suppose that  $d(\top, a) = n$  and  $d(\top, b) = m$ . We are interested in the following probability:

$$\begin{aligned} P(m | n) &:= P(b \in S(\top, m) | a \in S(\top, n)) \\ &= \sum_{r=0}^l P(m | r, n) P(r | n) \end{aligned} \quad (4)$$

where the following notation was used

$$\begin{aligned} P(m \mid r, n) &:= P(b \in S(\top, m) \mid b \in S(a, r), a \in S(\top, n)) \\ P(r \mid n) &:= P(b \in S(a, r) \mid a \in S(\top, n)) \end{aligned}$$

If mutation radius  $r$  can be controlled via parameter  $\mu$ , then transition probability (4) depends on this parameter as well. Specific expressions for  $P_\mu(m \mid n)$  depend on the topology of  $\Omega$ . Let us consider the Hamming space.

Let  $\Omega$  be a space  $\mathcal{H}_\alpha^l := \{1, \dots, \alpha\}^l$  — a space of sequences of length  $l$  and  $\alpha$  letters and equipped with the Hamming metric  $d(a, b) := |\{i : a_i \neq b_i\}|$ . Then, given probability of mutation  $\mu(n) \in [0, 1]$  of each letter in the parent sequence  $a \in S(\top, n)$ , the probability that  $b \in S(a, r)$  is

$$P_\mu(r \mid n) = \binom{l}{r} \mu(n)^r (1 - \mu(n))^{l-r} \quad (5)$$

Probability  $P(m \mid r, n)$  is defined by the number of elements in the intersection of spheres  $S(\top, m)$  and  $S(a, r)$ :

$$P(m \mid r, n) = \frac{|S(\top, m) \cap S(a, r)|_{d(\top, a)=n}}{|S(a, r)|} \quad (6)$$

where cardinality of the intersection  $S(\top, m) \cap S(a, r)$  with condition  $d(\top, a) = n$  is computed as follows

$$\begin{aligned} |S(\top, m) \cap S(a, r)|_{d(\top, a)=n} = \\ \sum (\alpha - 2)^{r_0} \binom{n - r_-}{r_0} (\alpha - 1)^{r_+} \binom{l - n}{r_+} \binom{n}{r_-} \end{aligned} \quad (7)$$

where the triple summation runs over  $r_0, r_+$  and  $r_-$  satisfying  $r_+ \in [0, (r + m - n)/2]$ ,  $r_- \in [0, (n - |r - m|)/2]$ ,  $r_- - r_+ = n - \max\{r, m\}$  and  $r_0 + r_+ + r_- = \min\{r, m\}$ . These conditions are based on metric inequalities for  $r, m$  and  $n$  (e.g.  $|n - m| \leq r \leq n + m$ ). The number of sequences in  $S(a, r) \subset \mathcal{H}_\alpha^l$  is

$$|S(a, r)| = (\alpha - 1)^r \binom{l}{r} \quad (8)$$

Substituting equations (5)–(8) into (4) we obtain the expression for  $P_\mu(m \mid n)$  in Hamming space  $\mathcal{H}_\alpha^l$ .

### Analytical Solutions for Special Cases

If fitness  $f$  is isomorphic to the Hamming metric, then transition probabilities  $P_\mu(x_{s+1} \mid x_s)$  are completely defined by  $P_\mu(m \mid n)$  with  $x_{s+1} = -m$  and  $x_s = -n$ . The corresponding Markov operator  $T_{\mu(n)}$  is then an  $(l+1) \times (l+1)$  matrix completely defining the evolution on  $[s, s+t]$ ,  $t \leq \lambda$ , for a given mutation rate function  $\mu(n)$ , if all individuals are allowed to reproduce (with selection, one has to compose  $T_{\mu(n)}$  with a selection operator). For example, one can show that for  $\lambda = 1$ , the optimal mutation rate is a step function:

$$\mu_1(n) := \begin{cases} 0 & \text{if } n < l(1 - 1/\alpha) \\ \frac{1}{2} & \text{if } n = l(1 - 1/\alpha) \\ 1 & \text{otherwise} \end{cases}$$

Unfortunately, analytical or numerical solutions to optimisation problems (2) or (3) are not available or tractable for  $\lambda > 1$  and large  $l$ . However, analysis allows us to derive some main features of an optimal control function  $\bar{\mu}(n)$ .

Minimisation of the convergence time to state  $m = 0$  is related to maximisation of probability  $P_\mu(m = 0 \mid n)$ . Because  $r = n$  and  $|S(\top, 0) \cap S(a, n)|_{d(\top, a)=n} = 1$ , it has the following expression:

$$P_\mu(m = 0 \mid n) = (\alpha - 1)^{-n} \mu^n (1 - \mu)^{l-n} \quad (9)$$

Mutation rate maximising this probability is obtained by taking its derivative  $P'_\mu$  over  $\mu$  to zero, and together with condition  $P''_\mu \leq 0$ , this gives  $n - l\mu = 0$  or

$$\mu_2(n) = \frac{n}{l} \quad (10)$$

This linear mutation control function has very intuitive interpretation — if sequence  $a$  has  $n$  letters different from the optimal sequence  $\top$ , then substitute  $n$  letters in the offspring. One can show that the linear function (10) is optimal for two-valued fitness landscapes with one optimal sequence, such as the Needle in a Haystack discussed in Example 1. This is because expected fitness  $\mathbb{E}_{\mu(x)}\{x_{s+t}\}$  in this case is completely defined by probability (9). For other fitness landscapes that are monotonic relative to the Hamming metric, function (10) is an approximation of the optimal control, because it does not take into account transition probabilities  $P_\mu(m \neq 0 \mid n \neq 0)$  between other (transient) states, which may influence the expected time of convergence to  $m = 0$ . As a result, the convergence can be very poor in the initial stages of evolution on  $[s, s+t]$ .

Bäck (1993) derived probability  $P_\mu(m < n \mid n)$  of ‘success’ in the space  $\mathcal{H}_2^l$  of binary sequences, and then considered mutation rates  $\hat{\mu}$  maximising its value for each  $n = d(\top, \omega)$ . Our equations (4)–(8) allow us to perform such optimisation for arbitrary  $\alpha$ . This method makes significant improvement over the linear control for the speed of convergence in the initial stages of evolution on  $[s, s+t]$ . We note, however, that the resulting mutation controls do not achieve optimal values (2) or (3). One can show that maximisation of  $P_\mu(m < n \mid n)$  is equivalent to maximisation of conditional expectation  $\mathbb{E}\{u(m, n) \mid n\} = \sum_m u(m, n) P_\mu(m \mid n)$  of a two-valued utility function:  $u(m, n) = 1$  if  $m < n$ ; 0 otherwise. This function has only two values, and such optimisation of  $\mu(n)$  is not precise for fitness functions with more than two values. In fact, analysis using absorbing Markov chains shows that linear control (10) achieves shorter expected times of convergence into absorbing state  $m = 0$ .

### Empirically Informed Mutation Rate

Another approach to optimal control of parameters in evolutionary systems is based on theories of information and optimal coding. In brief, one can reformulate problems (2)

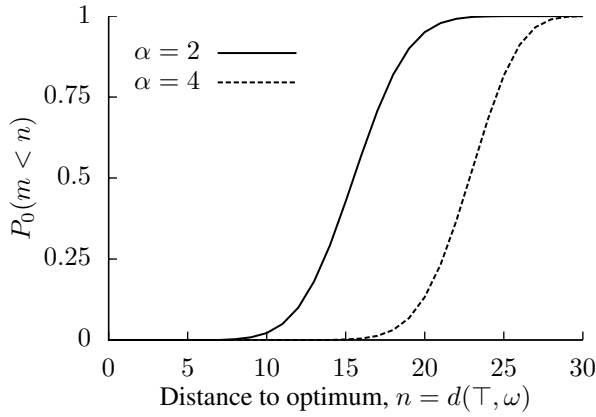


Figure 1: Cumulative distribution functions  $P_0(m < n)$  of distances to optimum under random distribution of sequences in  $\mathcal{H}_2^{30}$  and  $\mathcal{H}_4^{30}$ .

and (3) by replacing the time constraint  $t \leq \lambda$  with a constraint on information ‘distance’  $\mathbb{E}_{s+t}\{\ln(p_{s+t}/p_s)\} \leq \lambda$  of distribution  $p_{s+t} = T^t p_s$  from  $p_s$ . Although minimisation of information distance is not equivalent to minimisation of convergence time, this formulation has the advantage that the corresponding optimal values can be computed exactly and used to evaluate various control functions.

Our evaluation shows that adaptation  $\mathbb{E}\{x_{s+t}\} \geq v$  with the least information distance of  $p_{s+t}$  from  $p_s$  is achieved if mutation rate is identified with the CDF of the ‘least informed’ distribution  $P_0(x)$  of fitness values. In particular, assuming a uniform distribution  $P_0(\omega) = \alpha^{-l}$  of sequences in  $\mathcal{H}_\alpha^l$ , the distribution  $P_0(n) := P_0(\omega \in S(\top, n))$  of their distances from  $\top$  can be obtained by counting sequences in the spheres  $S(\top, n) \subset \mathcal{H}_\alpha^l$ . One can show also that this corresponds to binomial distribution with  $\mu = 1 - 1/\alpha$ :

$$P_0(n) = \binom{l}{n} \left(1 - \frac{1}{\alpha}\right)^n \left(\frac{1}{\alpha}\right)^{l-n} = \binom{l}{n} \frac{(\alpha - 1)^n}{\alpha^l}$$

In this case,  $\mathbb{E}\{n\} = l\mu = l(1 - 1/\alpha)$ . Under the minimal information distance assumption, the offspring will have very similar distribution, and the probability  $P_0(m < n)$  that an offspring is closer to  $\top$  is given by the CDF of  $P_0(n)$ , which can be used to control the mutation rate:

$$\mu_0(n) = P_0(m < n) = \sum_{m=0}^{n-1} P_0(m) \quad (11)$$

This mutation control function has the following interpretation — if sequence  $a$  has  $n$  letters different from the optimal sequence  $\top$ , then substitute each letter in the offspring with the ‘least informed’ probability of improvement relative to the current value  $n = d(\top, a)$ . Figure 1 shows  $P_0(m < n)$  for  $\mathcal{H}_2^{30}$  and  $\mathcal{H}_4^{30}$ . We note that minimisation of information distance of  $p_{s+t}$  from  $p_s := P_0$  corresponds to maximisation of entropy of  $p_{s+t}$ , but adaptation  $\mathbb{E}\{x_{s+t}\} \geq \mathbb{E}\{x_s\}$

leads to increasing the distance and decreasing the entropy (i.e. slow ‘cooling’ as in simulated annealing).

In the next section, we present nearly optimal mutation rate functions, obtained experimentally, and find that they correspond to CDFs of distributions that are skewed towards the optimum compared to the CDFs of the least informed distributions  $P_0$  (i.e. skewed to the left compared to those used in Figure 1). This can be explained by the fact that the offspring sequences do not have a uniform distribution in  $\mathcal{H}_\alpha^l$  during long intervals  $[s, s + t]$  due to adaptation  $\mathbb{E}_{\mu(n)}\{m\} \leq \mathbb{E}\{n\} = l(1 - 1/\alpha)$ . Therefore, the probabilities of improvement relative to the current fitness are higher than  $P_0(m < n)$ , and they can be approximated by empirical functions  $P_e(m < n)$ , observed during  $[s, s + t]$ . Thus, we refer to such a control as ‘informed’.

Finally, we note that if fitness is monotonic relative to the Hamming metric, then function  $P_e(m < n)$  can be replaced by function  $P_e(x_r > x)$  for fitness values. We conjecture that the corresponding control (1) of mutation rate should achieve good performance also in landscapes that are only weakly or  $\varepsilon$ -monotonic. Our experiments with an aptamer landscape (Rowe et al., 2010) support this hypothesis.

## Evolving Optimal Mutation Rates

To evaluate our theoretically derived mutation control functions, we have evolved such functions independently using a meta-genetic algorithm (Meta-GA). Populations of the Meta-GA comprised individual functions  $\mu(x)$ , which were then used to control mutation rates of another GA, referred to as Inner-GA. We first give some details about the Inner- and Meta-GAs, and then describe results of the experiments.

### Inner-GA

The Inner-GA is a simple generational genetic algorithm that uses no selection and no recombination. Each genotype in the Inner-GA is a sequence  $\omega \in \mathcal{H}_\alpha^l$ , and we used populations of 100 individuals. The initial population had equal numbers of individuals at each fitness value, and all runs within the same Meta-GA generation were seeded with the same initial population. Individuals were evolved by the Inner-GA for  $t = 500$  generations using simple mutation. The objective was to maximise a fixed fitness function  $f(\omega)$ . Here, we report results of the following three experiments:

1.  $\mathcal{H}_2^{30}$  (i.e.  $\alpha = 2, l = 30$ ) and fitness  $f(\omega) = -d(\top, \omega)$ , where  $d$  is Hamming metric.
2.  $\mathcal{H}_4^{10}$  (i.e.  $\alpha = 4, l = 10$ ) and fitness  $f(\omega) = -d(\top, \omega)$ , where  $d$  is Hamming metric.
3.  $\mathcal{H}_4^{10}$  (i.e.  $\alpha = 4, l = 10$ ) and fitness  $f(\omega)$  defined by a complete DNA-protein affinity landscape for 10-base-pair sequences (Rowe et al., 2010), which we refer to as the aptamer landscape.

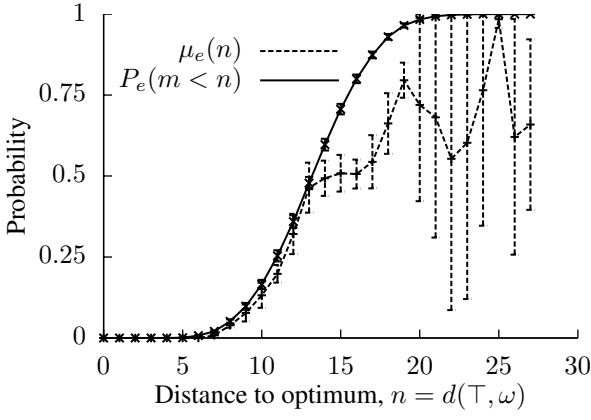


Figure 2: Average of evolved mutation functions  $\mu_e(n)$  and CDF  $P_e(m < n)$  for fitness  $f(\omega) = -d(\top, \omega)$  in  $\mathcal{H}_2^{30}$ .

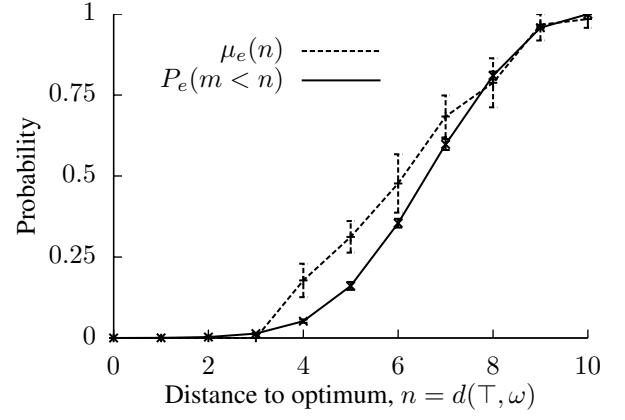


Figure 3: Average of evolved mutation functions  $\mu_e(n)$  and CDF  $P_e(m < n)$  for fitness  $f(\omega) = -d(\top, \omega)$  in  $\mathcal{H}_4^{10}$ .

### Meta-GA

The Meta-GA is a simple generational genetic algorithm that uses tournament selection (a good choice when little is known or assumed about the structure of the landscape). Each genotype in the Meta-GA is a mutation rate function  $\mu(x)$ , which is a sequence of  $l + 1$  real values  $\mu \in [0, 1]$  representing per-locus probabilities of mutation. We used populations of 100 individual functions, which were initialised to  $\mu(x) = 0$ .

The Meta-GA evolved functions  $\mu_e(x)$  for  $t = 5 \cdot 10^5$  generations to maximise the average fitness in the final generation of the Inner-GA. The Meta-GA used the following selection, recombination and mutation:

- Randomly select three individuals from the population and replace the least fit of these with a mutated crossover of the other two; repeat until all individuals from the population have been selected.
- Crossover (recombination) uses a single cut point chosen randomly (excluding the possibility of being at either end, so that there are no clones).
- Mutation adds a uniform-random number  $\Delta\mu \in [-.1, .1]$  to one randomly selected value  $\mu$  (mutation rate) on the individual (mutation rate function), but then bounds that value to be within  $[0, 1]$ .

The Meta-GA returned the fittest mutation rate function  $\mu_e(x)$ . In addition, we recorded empirical frequencies  $P_e(x)$  of fitness values  $x = f(\omega)$ , observed during running the Inner-GA for  $t$  generations on the relevant landscape and using that mutation rate function. We note that empirical frequencies  $P_e(x)$  counted only the number of phenotypic mutations (i.e. genetic mutations that result in a change in fitness). Empirical frequencies  $P_e(x)$  were then used to compute the cumulative distribution functions  $P_e(x_r > x)$ , which we then compared to the evolved  $\mu_e(x)$ .

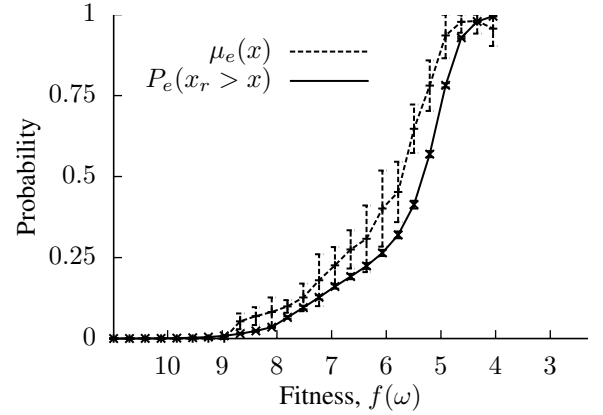


Figure 4: Average of evolved mutation functions  $\mu_e(x)$  and CDF  $P_e(x_r > x)$  for fitness  $f(\omega) = x$  from the aptamer landscape (Rowe et al., 2010) in  $\mathcal{H}_4^{10}$ .

### Experimental Results

We performed multiple runs of each experiment collecting multiple versions of evolved mutation control functions  $\mu_e(x)$  and cumulative distribution functions  $P_e(x_r > x)$  of observed fitness values. Figures 2, 3 and 4 show the average of these functions from 20 runs together with standard deviations. Figures 2 and 3 are for the experiments in  $\mathcal{H}_2^{30}$  and  $\mathcal{H}_4^{10}$  respectively, and with fitness  $f(\omega)$  defined by the negative Hamming distance  $-d(\top, \omega)$  to a fixed optimum  $\top$ . Figure 4 is for the experiment in  $\mathcal{H}_4^{10}$ , but with fitness  $f(\omega)$  defined by the complete aptamer landscape from (Rowe et al., 2010). The evolved functions  $\mu_e(x)$  are approximated fairly by the cumulative distribution functions  $P_e(x_r > x)$ , supporting heuristic (1). The mismatch in the areas of low fitness can be explained by slower convergence of functions  $\mu_e(x)$  in this part of the space  $\mathcal{H}_\alpha^l$  due to limited exploration of it by populations of 100 individuals in the Inner-GA, which are small relative to  $|\mathcal{H}_\alpha^l| = \alpha^l$ .

## Discussion

In this work, we have made some progress towards understanding optimal control of mutation rate, and some general principles can be formulated. It appears that choice of a representation space and its topology is crucial, as it defines the monotonic property of a fitness landscape. Our analysis was performed for a Hamming space, but the ideas can be extended to other spaces, such as a space of variable or infinite sequences with  $p$ -adic metric. If the right representation has been found, then specific formulae can be derived using geometric analysis in the representation space. These principles can also be extended to sexual reproduction and control of recombination. Our analysis and experiments suggest that an optimal control of mutation rate is based on statistical information about the distribution of fitness values.

The existence of optimal mutation rates that vary depending upon an individual's fitness raises a number of questions about the existence and control of variable mutation rates in biological organisms. For mutation rate control to have evolved in nature, a first prerequisite is that biological mutation rates can vary and are not simply minimised. There is ample evidence that mutation rates do vary in nature, between distantly (Drake et al., 1998) and closely-related (Matic et al., 1997) organisms, between regions of genomes (Lang and Murray, 2008) and even within an organism in stressful versus benign environments (Bjedov et al., 2003). However, the question of whether there may be an adaptive trait, allowing an individual organism to affect the number of mutations between itself and its offspring, dependent upon environmental cues, remains an open question. This would be an example of 'higher-order' selection, that is selection not on the immediate fitness of an individual, but on its ability to produce fitter descendants, potentially many generations later. Such higher order effects have always been questioned in biology, since they might be expected, in real populations, to be swamped by direct selective effects (Pigliucci, 2008). However, discussion has intensified recently over the concepts of 'robustness' and 'evolvability' (Masel and Trotter, 2010). These are higher order effects of somewhat unclear definition; the latter potentially relates directly to the control of mutation rate considered here. Very recent results from experimental evolution of microbial populations show that higher order evolvability effects can indeed play an important part in the evolution of real biological populations (Woods et al., 2011). However, in mechanistic terms, only the gross evolution of mutation rate itself (rather than mutation rate control) in 'mutator' strains has been identified in such experiments (Arjan et al., 1999).

If one moves from complete organisms to viruses and *in silico* quasi-biological evolution, there is more work on optimal mutation rates and their evolution. Optimal mutation rates can be identified (Kamp et al., 2002), relating to the concept of an 'error threshold' (Ochoa et al., 1999) the mutation rate at which selection can no longer be sufficient to

balance the deleterious effects of mutation (Biebricher and Eigen, 2005). However, Clune et al. (2008) used digital organisms to show that natural selection does not always effectively evolve optimal mutation rates for adaptation in the long-term, and this fact is particularly apparent when evolution occurs on a rugged fitness landscape. There is evidence that, in nature, epistasis is widespread (e.g. Costanzo et al., 2010), leading to rugged fitness landscapes. This potentially reduces the biological relevance of work, such as ours, with simple fitness functions. Nonetheless, even in rugged landscapes, biological evolution is, empirically, able to occur via locally monotonic accessible paths (Poelwijk et al., 2007), and we find a good agreement between the evolved and theoretical functions, even for a fitness landscape known to be rugged (e.g. Fig. 4 in Rowe et al., 2010). Similarly, temporal variation in fitness landscapes has been highlighted as biologically important (Costanzo et al., 2010), which, while it calls into question the biological relevance of optimal mutation rates in static landscapes, leads back to the potential biological importance of mutation rate variation in response to environmental cues (Stich et al., 2010).

Finally, we observe that understanding of evolution and dynamical systems, such as populations of organisms, may be facilitated by theories of information and information dynamics. In particular, optimisation problems, defined by functions (2) and (3), can be reformulated by replacing time with an information distance between probability distributions. Analytical solutions for such problems can be obtained (e.g. Belavkin, 2010), providing an alternative way to evaluate control functions. Although we do not report such evaluation here, we have observed that these information-theoretic optimal values are achieved when the mutation rate corresponds to a CDF of the 'least informed' distribution of fitness values. Understanding this relation between mutation rate control and information, along with its biological relevance, are some of the directions of our future work.

## Acknowledgements

This work was supported by EPSRC grant EP/H031936/1.

## References

- Ackley, D. H. (1987). An empirical study of bit vector function optimization. In Davis, L., editor, *Genetic Algorithms and Simulated Annealing*, chapter 13, pages 170–204. Pitman.
- Arjan, J. A., Visser, M., Zeyl, C. W., Gerrish, P. J., Blanchard, J. L., and Lenski, R. E. (1999). Diminishing returns from mutation supply rate in asexual populations. *Science*, 283(5400):404–6.
- Bäck, T. (1993). Optimal mutation rates in genetic search. In Forrest, S., editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 2–8. Morgan Kaufmann.
- Belavkin, R. V. (2010). Information trajectory of optimal learning. In Hirsch, M. J., Pardalos, P. M., and Murphey, R., editors, *Dynamics of Information Systems: Theory and Applications*,



volume 40 of *Springer Optimization and Its Applications Series*. Springer.

- Biebricher, C. K. and Eigen, M. (2005). The error threshold. *Virus Res*, 107(2):117–27.
- Bjedov, I., Tenaillon, O., Gerard, B., Souza, V., Denamur, E., Radman, M., Taddei, F., and Matic, I. (2003). Stress-induced mutagenesis in bacteria. *Science*, 300(5624):1404–9.
- Cervantes, J. and Stephens, C. R. (2006). “optimal” mutation rates for genetic search. In Cattolico, M., editor, *Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2006)*, pages 1313–1320, Seattle, Washington, USA. ACM.
- Clune, J., Misevic, D., Ofria, C., Lenski, R. E., Elena, S. F., and Sanjuan, R. (2008). Natural selection fails to optimize mutation rates for long-term adaptation on rugged fitness landscapes. *PLoS Comput Biol*, 4(9):e1000187.
- Costanzo, M., Baryshnikova, A., Bellay, J., Kim, Y., Spear, E. D., et al. (2010). The genetic landscape of a cell. *Science*, 327(5964):425–31.
- Cowperthwaite, M. C. and Meyers, L. A. (2007). How mutational networks shape evolution: Lessons from rna models. *Annual review of Ecology, Evolution and Systematics*, 38:203–230.
- Cox, M. M., Keck, J. L., and Battista, J. R. (2010). Rising from the ashes: DNA repair in *Deinococcus radiodurans*. *PLoS Genet*, 6(1):e1000815.
- Drake, J. W., Charlesworth, B., Charlesworth, D., and Crow, J. F. (1998). Rates of spontaneous mutation. *Genetics*, 148(4):1667–86.
- Eiben, A. E., Hinterding, R., and Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124–141.
- Eigen, M., McCaskill, J., and Schuster, P. (1988). Molecular quasispecies. *Journal of Physical Chemistry*, 92:6881–6891.
- Falco, I. D., Cioppa, A. D., and Tarantino, E. (2002). Mutation-based genetic algorithm: performance evaluation. *Applied Soft Computing*, 1(4):285–299.
- Fisher, R. A. (1930). *The Genetical Theory of Natural Selection*. Oxford University Press, Oxford.
- Fogarty, T. C. (1989). Varying the probability of mutation in the genetic algorithm. In Schaffer, J. D., editor, *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 104–109. Morgan Kaufmann.
- Hakem, R. (2008). DNA-damage repair; the good, the bad, and the ugly. *Embo J*, 27(4):589–605.
- Kamp, C., Wilke, C. O., Adami, C., and Bornholdt (2002). Viral evolution under the pressure of an adaptive immune system: Optimal mutation rates for viral escape. *Complexity*, 8(2):28–33.
- Knight, C. G., Platt, M., Rowe, W., Wedge, D. C., Khan, F., Day, P. J., McShea, A., Knowles, J., and Kell, D. B. (2009). Array-based evolution of DNA aptamers allows modelling of an explicit sequence-fitness landscape. *Nucleic Acids Res*, 37(1):e6.
- Lang, G. I. and Murray, A. W. (2008). Estimating the per-base-pair mutation rate in the yeast *saccharomyces cerevisiae*. *Genetics*, 178(1):67–82.
- Masel, J. and Trotter, M. V. (2010). Robustness and evolvability. *Trends Genet*, 26(9):406–14.
- Matic, I., Radman, M., Taddei, F., Picard, B., Doit, C., Bingen, E., Denamur, E., and Elion, J. (1997). Highly variable mutation rates in commensal and pathogenic *Escherichia coli*. *Science*, 277(5333):1833–4.
- Mühlenbein, H. (1992). How genetic algorithms really work: Mutation and hillclimbing. In einhard Männer and Manderick, B., editors, *Parallel Problem Solving from Nature 2*, pages 15–26, Brussels, Belgium. Elsevier.
- Nix, A. E. and Vose, M. D. (1992). Modeling genetic algorithms with Markov chains. *Annals of Mathematics and Artificial Intelligence*, 5(1):77–88.
- Ochoa, G. (2002). Setting the mutation rate: Scope and limitations of the 1/l heuristics. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2002)*, pages 315–322, San Francisco, CA. Morgan Kaufmann.
- Ochoa, G., Harvey, I., and Buxton, H. (1999). Error thresholds and their relation to optimal mutation rates. In *Proceedings of the Fifth European Conference on Artificial Life (ECAL’99)*, volume 1674 of *Lecture Notes in Artificial Intelligence*, pages 54–63, Berlin. Springer-Verlag.
- Orr, H. A. (2005). The genetic theory of adaptation: a brief history. *Nat Rev Genet*, 6(2):119–27.
- Pigliucci, M. (2008). Is evolvability evolvable? *Nat Rev Genet*, 9(1):75–82.
- Poelwijk, F. J., Kiviet, D. J., Weinreich, D. M., and Tans, S. J. (2007). Empirical fitness landscapes reveal accessible evolutionary paths. *Nature*, 445(7126):383–6.
- Rowe, W., Platt, M., Wedge, D. C., Day, P. J., and Kell, D. B. (2010). Analysis of a complete DNA-protein affinity landscape. *Journal of Royal Society Interface*, 7(44):397–408.
- Stich, M., Manrubia, S. C., and Lazaro, E. (2010). Variable mutation rates as an adaptive strategy in replicator populations. *PLoS One*, 5(6):e11186.
- Vafaei, F., Turán, G., and Nelson, P. C. (2010). Optimizing genetic operator rates using a Markov chain model of genetic algorithms. pages 721–728. ACM.
- Waddington, C. H. (1957). *The strategy of the genes; a discussion of some aspects of theoretical biology*. Allen & Unwin, London.
- Woods, R. J., Barrick, J. E., Cooper, T. F., Shrestha, U., Kauth, M. R., and Lenski, R. E. (2011). Second-order selection for evolvability in a large *Escherichia coli* population. *Science*, 331(6023):1433–6.
- Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding and selection in evolution. *Proceedings of the Sixth International Congress of Genetics*, 1:356–366.
- Yanagiya, M. (1993). A simple mutation-dependent genetic algorithm. In Forrest, S., editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, page 659. Morgan Kaufmann.

# Staging the Self-Assembly Process Using Morphological Information

Navneet Bhalla<sup>1</sup>, Peter J. Bentley<sup>2</sup>, Peter D. Vize<sup>1</sup> and Christian Jacob<sup>1</sup>

<sup>1</sup>University of Calgary, Calgary, AB, Canada T2N 1N4

<sup>2</sup>University College London, London, UK WC1E 6BT  
nbhalla@ucalgary.ca

## Abstract

One of the practical challenges facing the creation of self-assembling systems is being able to exploit a limited set of fixed components and their bonding mechanisms. Staging addresses this challenge by dividing the self-assembly process into time intervals, and encodes the construction of a target structure in the staging algorithm itself and not exclusively into the design of components. Previous staging strategies do not consider the interplay between component physical features (morphological information). In this work we use morphological information to stage the self-assembly process, with the benefit of reducing assembly errors and leveraging bonding mechanism with rotational properties. Four experiments are presented, which use heterogeneous, passive, mechanical components that are fabricated using rapid prototyping. Two orbital shaking environments are used to provide energy to the components, and to investigate the role of morphological information with component movement in either two or three spatial dimensions. The experiments demonstrate, as proof-of-concept, that staging enables the self-assembly of more complex morphologies not otherwise possible.

## Introduction

Comprehending the principles of self-assembly has been described as one of the important aspects to understanding life (Ingber, 1998). Self-assembly is also considered to being an *enabling technology* for the creation of artificial systems (Pelesko, 2007). Constructing systems with natural characteristics (e.g. self-assembly, self-repair, and parallel construction) as a form of emergent engineering requires an understanding of the interplay between programmability/controllability and self-organisation (Doursat, 2008).

One important challenge when creating artificial self-assembling systems is caused by the use of components that lack the plasticity of biological cells. Using components that cannot differentiate results in self-assembly being constrained to a limited set of fixed components and their bonding mechanisms (Demaine et al., 2008). One strategy to address this challenge is to divide the self-assembly process into stages, referred to as *staged* or *hierarchical* self-assembly. Demaine et al. (2008) formalised the method of staging where components can be added to, or removed from, an environment at various time intervals.

Demaine et al. (2008) demonstrated the benefits of staging theoretically using abstract tiles, where staging the self-assembly process was based on the temporal aspects of conducting laboratory experiments. In contrast, we use physical components, and propose using morphological information as the dividing basis to staging the self-assembly process, inspired by biological development. Here we consider how physical features in a set of heterogeneous, passive, mechanical components can be exploited to reduce potential assembly errors, leverage rotational bonding mechanisms, and create structures with symmetrical/asymmetrical features. Our staging strategy is consistent with the definition of self-assembly (Whitesides and Gryzbowski, 2002), as a process involving components that can be controlled through their proper design and their environment, and where components can adjust their relative positions.

Staged self-assembly provides the advantage of encoding the construction of a target structure in the staging algorithm itself and not exclusively into the design of the components. For example, a staging algorithm can be used to reintroduce previously used components and bonding mechanisms at later time intervals, prevent the formation of holes, and create more complex morphologies that may not be otherwise possible due to shape conflicts between components.

The following section provides background material to which our staging strategy is built upon. Next, an overview of our approach is provided, including a theoretical model and physical description of the components and environments used. Four experiments follow that demonstrate the creation of self-assembled structures, from a set of components that are divided into two time intervals based on their physical features. Components are fabricated using rapid prototyping, and are placed in one of two orbital shaking environments (on a tray surface or in a jar of fluid). These two environments are used to demonstrate the role of morphological information in terms of component movement spatially in two and three dimensions (2D and 3D). We conclude by summarising how this work provides proof-of-concept evidence for staging the self-assembly process using morphological information.

## Background

Biological development utilises explicit stages in its provision of a solution to the construction of multicellular organisms (Wolpert, 1998). The explicit stages in biological development are often irreversible, and cannot be repeated at later stages, such as invagination, gastrulation, and the formation of a body plan. Staged development in nature allows for the creation of more complex phenotypes, which otherwise would not be possible (Wolpert, 1998).

A challenge towards the creation of self-assembling systems is the use of fixed components in contrast to components that can differentiate and communicate (e.g. cells in biological organisms). *DNA nanotechnology* is one example of an application area using fixed components, such as DNA tiles (using interwoven double-stranded DNA to create the body of a tile, and single DNA strands extending from the edges of a tile's body; Winfree et al., 1998). The staged Tile Assembly Model (sTAM) addresses this challenge by incorporating the temporal aspects of conducting laboratory experiments, using DNA tiles for example, into the self-assembly process (Demaine et al., 2008).

The sTAM is an extension to the abstract Tile Assembly Model (aTAM; Winfree, 1998). The aTAM was developed to provide a theoretical framework to investigate the assembly of square tiles (based on DNA tiles) in a square lattice environment. A tile type is defined by the bonding domains on the North, West, South, and East edges of a tile. At least one seed tile must be specified to start the self-assembly process. Tiles cannot be rotated or reflected. There cannot be more than one tile type that can be used at an assembly location in the growing structure. Tile types are in infinite supply, of equal concentration, in the model. All tiles are added to the same environment, *one-pot-mixture*. Tiles can only bond together if the interactions between them meet or exceed the *temperature* parameter. As a result, temperature dictates *co-operative bonding*. The seed tile is first placed in the environment, and additional tiles are added one at a time if the bonding constraints are satisfied.

The sTAM extends the aTAM by dividing the self-assembly process into time intervals. Components can be added to, or removed from, as set of environments, mirroring the laboratory operations of adding/filtering DNA-based components to solutions that can be mixed together. The sTAM has been used to investigate the algorithmic construction of structures, such as a fully connected  $n \times n$  square ( $n \in \mathbb{N}$ ). The construction of a square is problematic, as assembling tiles must be coordinated to prevent the occurrence of holes. The sTAM has shown an algorithmic efficiency with minimal tile sets and bonding mechanisms (not requiring co-operative bonding, at temperature one) in the construction of such structures. This efficiency is due to staging, and is an advantage over the aTAM itself that relies on co-operative bonding (Rothmund and Winfree, 2000), or other extensions to the aTAM that use either changes in temperature

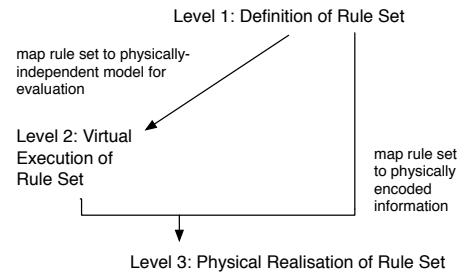


Figure 1: Three-level approach to self-assembly design.

(Kao and Schweller, 2006) or by varying the concentration of tiles (Adleman et al., 2001; Doty, 2009).

*Situated development* is another method investigating staged construction, where artificial evolution was used to evolve the assembly plan of a structure (Rieffel and Pollack, 2005). Based on rapid prototyping, assembly plans were evolved using *permanent* and *temporary* components which were “dropped” in an environment. Temporary components act as scaffolding and can be removed (representing how support material can be removed in rapid prototyping).

In contrast to Demaine et al. (2008) and Rieffel and Pollack (2005), physical examples of staged self-assembly include Wu et al. (2002) where templates were used to self-assemble spherical beads into substructures with specific patterns (e.g. linear, triangular, and hexagonal shapes). As well, He et al. (2008) used three-point start motif tiles to self-assemble tetrahedrons, dodecahedrons, and buckyballs by controlling the motif length and concentration of tiles in a two-step process. Despite this work, there is little (if any) literature that describes the use of morphological information to stage the self-assembly process.

## Staging and the Three-Level Approach

The three-level approach provides a high-level description to designing self-assembling systems via physically encoded information (Bhalla et al., 2010). The three levels include: (1) definition of rule set, (2) virtual execution of rule set, and (3) physical realisation of rule set (Fig. 1). Here we extend the three-level approach to incorporate our staging strategy. At level one, a new self-assembly rule is introduced to specify which components are present at a particular time interval. To accommodate this new rule, an extension to a self-assembly model based on the aTAM is provided at level two. Finally, physical features of components that are exploited in our staging experiments is described at level three.

### Level One: Definition of Rule Set

A system is described by three categories of self-assembly rules, *component*, *environment*, and *system*, which are in the context of component movement spatially in 2D or 3D.

Component rules specify shape and information. Conceptually similar to DNA tiles, components are either squares

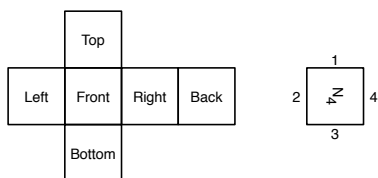


Figure 2: 3D component spatial relationship, and an example of information orientation on a 3D component's face.

(2D) or cubes (3D). Each edge/face of a component serves as an information location (Fig. 2), in either a four-point (Top-Left-Bottom-Right) or six-point arrangement (Top-Left-Bottom-Right-Front-Back). Information is represented by a capital letter (A to H for 2D components, and I to T for 3D components). A subscript (1 to 4) is used with each capital letter (e.g.  $N_4$ ) to indicate orientation on a 3D component's face. The dash symbol (—) represents a neutral site (where no assembly information is present). The spatial relationship of a component's information defines its type.

Environment rules specify environmental conditions such as temperature ( $\phi$ ) and *boundary* constraints. An assembly protocol must at least meet the temperature for assembly bonds to occur. The boundary confines components to the environment. Components are permitted to translate and rotate in 2D and 3D systems. In addition, components have rotational information and can be reflected in 3D systems.

System rules specify component type frequency in each time interval ( $\psi$ ), and two interaction rules (*fits* and *breaks*). Time intervals indicate when components are added to a single environment (e.g.  $\psi_0$ ; using a subscript 0 to n, where  $n \in \mathbb{N}$  and 0 indicates the start of the self-assembly process). If two complementary pieces of information come into contact, (e.g. A fits B), it will cause them to assemble. This rule type is commutative (e.g. if A fits B, then B fits A). Furthermore, fits rules encapsulate component-to-component rotational interactions in 3D systems. A subscript (360, 180, and 90) is used to represent if the faces of complementary 3D components can fit together in four, two, or in one way respectively (e.g. M fits<sub>180</sub> N). If two assembled pieces of information experience at least a temperature of two ( $\phi_2$ ), then their assembly breaks. The system rules in conjunction with their physical counterparts is provided at the end of this section, *Level Three: Physical Realisation of Rule Set*.

## Level Two: Virtual Execution of Rule Set

At level two, a self-assembly rule set is mapped to an abstract tile model for computational efficient evaluation, and is used to determine if physical evaluation of a self-assembly rule set is applicable at level three. We extend the concurrent Tile Assembly Model (cTAM; Bhalla et al., 2010) to incorporate staging. In contrast to the aTAM, the cTAM is better suited to the type of self-assembling systems used here by al-

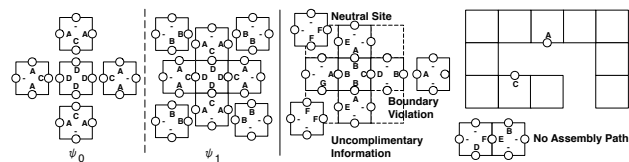


Figure 3: 2DscTAM example, and 2D assembly violations.

lowing multiple substructures to self-assemble concurrently, not using seed tiles, permitting more than one tile type to be used at an assembly location, and requiring all tiles to be in the same one-pot-mixture environment. The extended cTAM is referred to as the 2D and 3D staged concurrent Tile Assembly Model (2DscTAM and 3DscTAM). Components are permitted to translate and rotate in both the 2DscTAM and the 3DscTAM, but only be reflected in the 3DscTAM.

The input into the 2DscTAM and the 3DscTAM is the number of time intervals, and the multiset of components in each interval (type and frequency). At the start of each time interval, the components corresponding to the current time interval are added to the environment (Fig. 3). A single assembly operation is applied during a time interval, initialised by selecting a single tile/substructure with an open assembly location at random. If no other tile/substructure has an open complementary information location, then the location on the first tile/substructure is labelled *unmatchable*. If there are tiles/substructures with open complementary information locations, all those tiles/substructures are put in an *assembly candidate list*. From the assembly candidate list, tiles/substructures are selected at random until a tile/substructure can be added. If no such tile/substructure can be added, due to an *assembly violation* (Fig. 3), then the location is labelled *unmatchable*. If a tile/substructure can be added, the open assembly locations on the two tiles/substructures are updated and labelled *match* (all applicable assembly locations, including their rotational properties in the 3D case, must match when adding two substructures). This process repeats until all assembly locations are set to either match or unmatchable. At the end of a time interval, the resulting structures are placed in a single grid environment to determine if boundary violations occur. Before starting the next time interval, all unmatchable information locations are reset. The algorithm repeats, and halts when all time intervals have been completed in sequence.

An added constraint to the 3DscTAM is that substructures (with three or more components) cannot assemble together. This constraint represents observations in preliminary physical experiments conducted by the authors.

## Level Three: Physical Realisation of Rule Set

Components are physically realised using rapid prototyping, at level three. Both 2D and 3D components are defined by their design space (set of physically feasible designs, Fig. 4

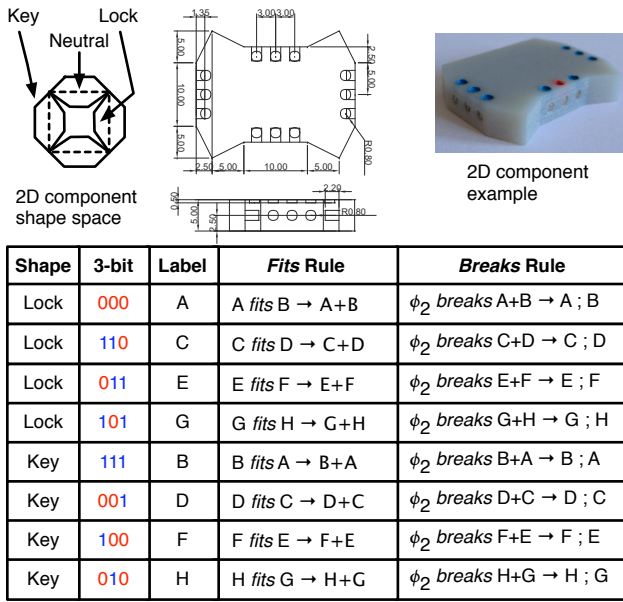


Figure 4: 2D component specification (construction units in mm), and 2D interaction rules (where red/zero and blue/one represent magnetic south and north respectively, and ' $\rightarrow$ ' transition, '+' assembly, and ';' disassembly).

and 5). The design space is a combination of a shape and an assembly protocol space. For both 2D and 3D components, a key-lock-neutral concept defines the shape space. A linear 3-magnetic-bit and a planar 5-magnetic-bit encoding scheme define the assembly protocol space for 2D and 3D components respectively. Magnets are placed within the edges or faces of 2D and 3D components respectively, and are not flush with a component's surface. The result of an air gap allows for adjustable component interactions and selective bonding (Whitesides and Gryzbowski, 2002). Although Miyashita et al. (2009) investigated how component shape and magnetic bonding affects the self-assembly process, they did not consider this morphological information in the context of staged self-assembly.

Here, lock-to-lock interactions can never occur due to their shape. This shape characteristic is influential in assigning 3-magnetic-bit and 5-magnetic-bit encodings to keys and locks. One magnet is placed in each position associated with a key, and two magnets are placed in each position associated with a lock. Strong bonding is ensured for key-to-lock interactions, and weak bonding between key-to-key interactions. The potential occurrence of weak bonding can be reduced with an appropriate physical temperature setting.

The four pairs of complimentary 3-magnetic-bit encodings can be optimally assigned to keys and locks to reduce assembly errors, as any key-to-lock error is at worst a one out of three match. Since this is not above a 50% match, bonding will not occur. Whereas the six pairs of unique

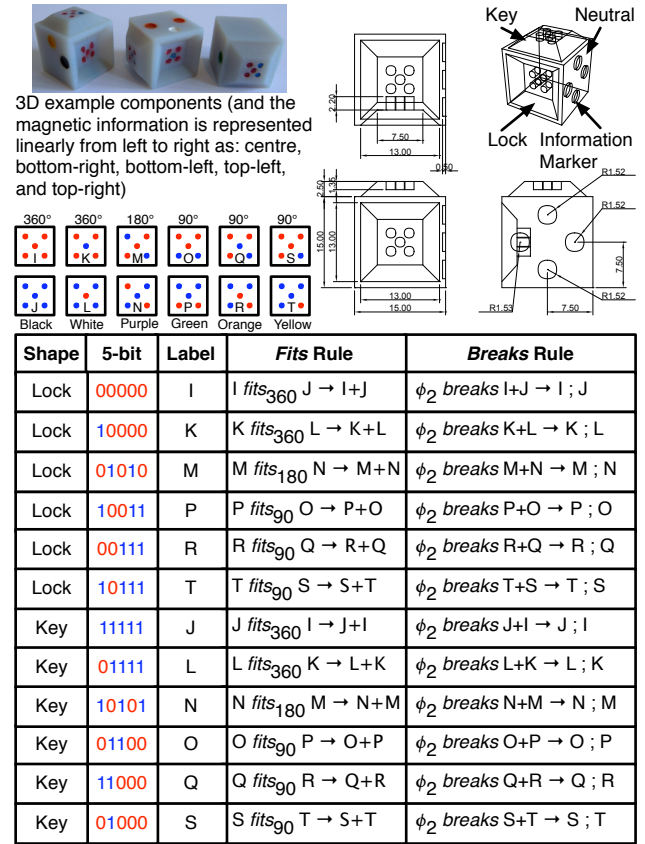


Figure 5: 3D component specification (construction units in mm), and 3D interaction rules (where red/zero and blue/one represent magnetic south and north respectively, and ' $\rightarrow$ ' transition, '+' assembly, and ';' disassembly).

complimentary pairs of 5-magnetic-bit encodings (accounting for planar rotation of a component's face) cannot be optimally assigned to keys and locks to reduce assembly errors. In this case, optimal assignment is considered with respect to which encodings are included to construct a target structure. It should be noted that these six encodings encapsulate rotational information for 3D component-to-component interactions, where two pairs encapsulate 360°, one pair encapsulates 180°, and three pairs encapsulate 90° rotational interactions. The 90° encodings have the potential for self-errors between complementary pairs, i.e. a three out of five match. A physical temperature to break three out of five matches, while maintaining five out of five matches, is strived for.

Orbital shakers form the environments for both 2D and 3D components. 2D components are placed on the surface of a tray, and a lid is used to prevent component reflections. 3D components are placed in a jar of mineral oil, to allow components to move freely in 3D space, and prevent oxidation affecting the magnets. The designs for both environments result from earlier experiments conducted by the authors.

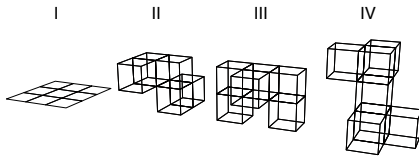


Figure 6: Four target structures for the experiments.

## Experiments and Results

We present four experiments that were conducted to test our approach to staging the self-assembly process using morphological information. The purpose of these experiments is to demonstrate, as proof-of-concept, that staging can enable the self-assembly of closed target structures not otherwise possible. Closed refers to structures with defined boundaries (Whitesides and Gryzbowski, 2002). A target structure was assigned to each experiment (one 2D and three 3D experiments, Fig. 6). Here, the self-assembly process is staged (divided) into two time intervals, where components are only added to a one-pot-mixture environment. Component physical features, such as key and lock shapes and magnetic-bit patterns, are morphological information.

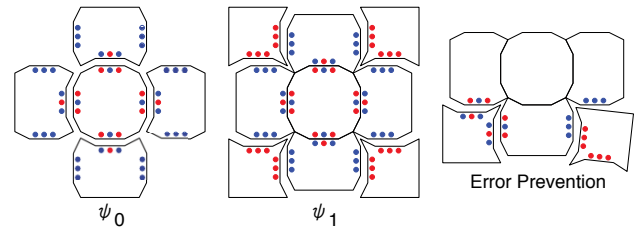
The independent variable is the use of two time intervals. The dependent variable is the resulting self-assembled structures. Enough components are supplied to create one 2D target structure and two 3D structures (due to boundary constraints of the environment). Ten trials are run for each experiment. A virtual trial (level two) is evaluated to being successful if all the potential number of target structures are achieved. A physical trial (level three) is evaluated to being successful if at least one target structure is achieved. The staging strategies and level one rules were designed by the authors. 2D and 3D experimental procedures and results are provided in terms of the three-level approach.

### Two-Dimensional System

The staging strategy for creating the 2D  $3 \times 3$  square target structure is to construct the *centre* and *edges* of the square in the first time interval, and construct the *corners* of the square in the second time interval (Fig. 7). In the first time interval, potential errors between the edge components can be reduced by appropriate selection of 3-magnetic-bit codes and the use of lock shapes to assemble to the centre component. The morphology of the substructure after the first time interval has corner features that can reduce assembly errors with the use of corner components that use only lock assembly shapes. The neutral edges of the corner components effectively block a corner component from assembling to the substructure in an improper orientation (Fig.7).

### 2D Level One Definition of Rule Set for Experiment

Fig.7 provides the component rules. The control group represents components that were not divided into time intervals



Target Structure	Staged Component Set
I	$\psi_0 \{1 \times (D,D,D,D), 4 \times (B,-,B,C)\}$ $\psi_1 \{4 \times (-,A,A,-)\}$

Figure 7: Staging strategy for target structure I, and error prevention due to shape and proper 3-magnetic-bit pattern selection (e.g. avoid magnetic repulsion configuration).

(non-staged). The experimental group used the same components, but divides them into two time intervals (staged). Interaction rules from Fig. 4 were applicable to both groups.

**2D Level Two Experimental Setup** The components from Fig.7 were mapped to an abstract representation for the 2DscTAM. Each component's shape was a unit square. The size of the environment was  $10 \times 10$  units (as a representation of width  $\times$  depth, and the ratio between component and environment size). A different random seed was used to initialise the 2DscTAM for each trial.

**2D Level Two Experimental Results** The staged components successfully created one target structure in each of the ten trials. None of the non-staged components were able to create one target structure. The unsuccessful non-staged trials either resulted in a set of substructures (due to edge and corner components assembling in incorrect orientations), or the creation of a  $3 \times 3$  open square. The results at level two were analysed using Fisher's Exact Test (one sided) for binary data (Cox and Snell, 1989). The results are statistically significant with a p-value of 0.

**2D Level Three Experimental Setup** A level three translation was preformed for both the staged and non-staged components (to observe the physical results of non-staged components). Components were mapped following Fig.7.

An Eden 333 Polyjet rapid prototyping machine was used to fabricate the components from Vero Grey resin. Neodymium (NdFeB) disc magnets ( $1/16'' \times 1/32''$ , diameter  $\times$  radius; grade N50) were inserted into the components. Blue/red paint (north/south) marked the magnets.

The environment size was mapped in accordance with the base component's size, to specify the dimensions of the circular tray environment. The tray was fabricated using a Dimensions Elite rapid prototyping machine, using ABS plas-



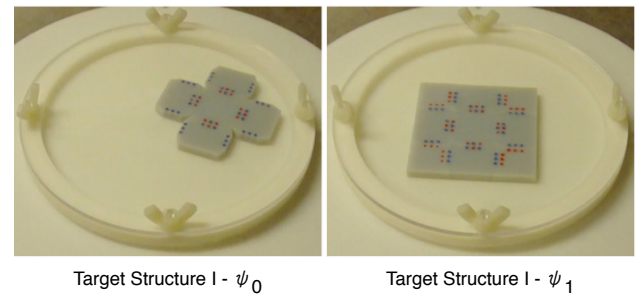
tic (sparse-fill option was used to create a rough surface texture). The outer radius of the tray is 135 mm and the inner radius is 125 mm, while the outer wall height is 9 mm and the inner wall height is 6 mm. The tray was mounted to a Maxi Mix II Vortex Mixer (using a tray mounting bracket, also fabricated using the Dimensions printer). A tray lid was cut using a Trotec Speedy 300 Laser Engraver laser cutting machine, using 2 mm clear acrylic sheet. The tray lid was secured to the tray using polycarbonate screws and wing nuts.

Each physical trial followed seven steps (Bhalla et al., 2010). (1) Set the speed control on the Maxi Mix II Vortex mixer to 1,050 rpm. This speed created an appropriate shaking level (environment temperature) to maintain fits rules, and to mostly break partially matched magnetic codes. (2) Secure the mixer to a table, using a 3" c-clamp and six hex nuts (to help secure the c-clamp to the back of the mixer). (3) Randomly place components on the surface of the tray (trying to ensure that complementary bonding sites on the components are not in-line with each other). (4) Secure the tray lid. (5) Run the mixer for 20 minutes for a non-staged trial, or for two 10 minute intervals for a staged trial. (6) Turn the mixer off. (7) Record the state of the system, observing: the number of target structures created, the number of matching errors (between conflicting physical information, where no fits rule is applicable), and the number of assembly errors (partial attachment between corresponding physical information, where a fits rule is applicable).

**2D Level Three Experimental Results** The level-three results are provided in Fig. 8, with an example of the end of each time interval of a successful trial. For both component groups, no matching and assembly errors were observed in the ten trials. Only partial structures were observed, and no open  $3 \times 3$  squares, were observed at the conclusion of the non-staged trials. Using Fisher's Exact Test, this experiment is statistically significant at the 0.01 level (i.e. there is a 99% certainty the results are not due to chance).

### Three-Dimensional Systems

The three 3D target structures have a three component common *core* structure, and vary in the number of *periphery* components (increasing from two, three, and four). The core structure requires two specialised  $90^\circ$  bonds, whereas the perimeter components only require general  $360^\circ$  bonds. As observed by the authors in preliminary 3D experiments, substructures consisting of at least three components are not able to assemble together. Given that the likelihood of general  $360^\circ$  bonds occurring is more likely than specialised  $90^\circ$  bonds, the staging strategy for creating the three 3D target structures is to construct the core substructure in the first time interval, and construct the periphery substructures in the second time interval (Fig. 9). The first time interval leverages the specialised component rotational information. Lock shapes for the  $360^\circ$  bonds are used as part of the mor-



Target Structure	Group	Successful	Unsuccessful
I	staged	7	3
	non-staged	0	10

Figure 8: Successful target structure I example trial, and the number of successful/unsuccessful 2D trials.

phology of the components in the first time interval, to reduce potential matching errors between specialised and general bonds. Furthermore, the resulting morphologies of the resulting core substructures at the end of the second time interval consist only of neutral and lock shapes, preventing assembly between the core substructures.

### 3D Level One Definition of Rule Set for Experiments

The component rules for the 3D experiments is provided in Fig. 9. Control groups and experimental groups represent non-staged and staged (using two time intervals) component sets respectively. The environment temperature was one. The interaction rules from Fig. 5 applied to both groups.

### 3D Level Two Experimental Setup

The components from Fig. 9 were mapped to an abstract representation for the 3DscTAM. A component's base shape was a unit cube. The size of the environment was  $4 \times 4 \times 4$  units (representing width  $\times$  depth  $\times$  height, and the ratio between component and environment size). A different random seed was used to initialise the 3DscTAM for each trial.

### 3D Level Two Experimental Results

The staged components, for each experiment, successfully created two target structures in each of the ten trials. Whereas, the non-sateged components were not able to create a target structure. As expected, the unsuccessful non-sateged components resulted in substructures consisting of three components (favouring assemblies with  $360^\circ$  bonds) or two components. The results at level two are statistically significant with a p-value of 0 using Fisher's Exact Test for binary data.

### 3D Level Three Experimental Setup

As with the 2D experiment, a level-three translation was performed for both staged and non-staged components (to observe the physical results of non-staged components). Components were

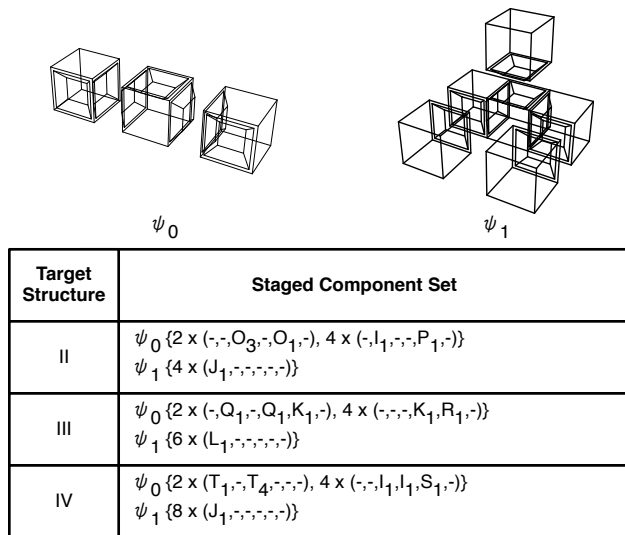


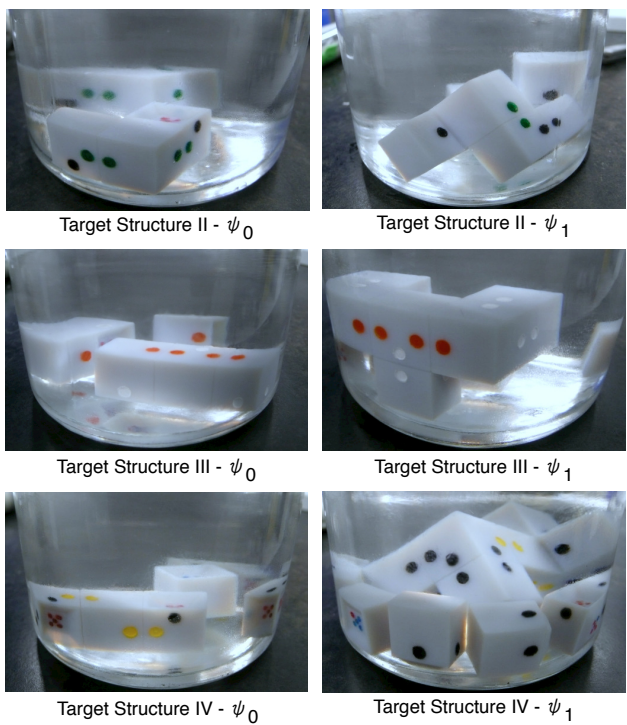
Figure 9: Staging strategy for target structure III (applicable to target structures II and IV).

mapped following Fig. 9, and were fabricated using a similar procedure as the 2D components (with the addition of colour paints to represent rotational information, Fig. 5).

500 mL clear glass, wide-mouth jars with rubber lined lids were used to contain components (91 mm×95 mm; diameter×height). A Trotec Speedy 300 Laser Engraver was used to construct the parts, using 3 mm acrylic sheet, for the jar rack. The rack was assembled using adhesive, screws, and hex nuts. The jar rack was placed on a New Brunswick Scientific Excella E1 Platform Shaker. 325 mL of Rogier Pharma light mineral oil was measured using a graduated cylinder, and poured into the jars (one for each experiment).

Each physical trial followed six steps. (1) Place three jars of mineral oil on the jar rack. (2) Randomly place the components for each experiment into the appropriate jar. (3) Secure the jar lids. (4) Turn the shaker on by setting the speed to 32.5 rpm. (5) Run the shaker for 40 minutes for a non-staged trial, or for two 20 minute intervals for a staged trial. (6) Record the state of each system, observing: the number of target structures created, the number of matching errors, the number of assembly errors, and the number of rotation errors (between complementary components).

**3D Level Three Experimental Results** The 3D level-three results are provided in Fig. 10, along with examples of the end of each time interval of a successful staged trial. For each experiment, no matching and assembly errors were observed in the ten trials. Rotational errors were observed in each staged experiment (Fig. 11). Using Fisher’s Exact Test, the first two 3D experiments are statistically significant at the 0.05 level and the third experiment was statistically significant at the 0.50 level (i.e. there is a 95% and 50% cer-



Target Structure	Group	Successful	Unsuccessful
II	staged	4	6
	non-staged	0	10
III	staged	5	5
	non-staged	0	10
IV	staged	1	9
	non-staged	0	10

Figure 10: Successful target structure II, III, and IV example trials, and the number of successful/unsuccessful 3D trials.

tainty the results are not due to chance). Even though one successful staged trial was observed with the third 3D experiment, we do not consider the result statistically relevant.

**Discussion** Four experiments were conducted to demonstrate our morphological information based staging strategy. At level two, all of the staged components sets were able to achieve their respective target structures, whereas none of the non-staged components were able to. All the staged component sets, except for the third 3D experiment, were able to successfully construct their respective target structures at a statistically significant level (with 99% and 95% confidence for the 2D and 3D experiments), at level three.

One physical target structure was achieved in the third 3D experiment, and we observed in the trials a *layering effect* of components/substructures that inhibited the self-assembly of this target structure (IV). As future work, we look to build neutrally buoyant components to address this issue.



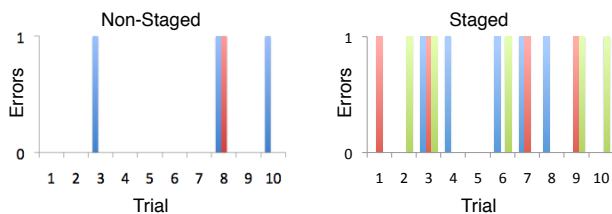


Figure 11: Rotational errors at the end of each 3D trial (target structures II blue, III red, and IV green).

We are also investigating the use of higher-order magnetic-bit codes, additional magnetic-bit patterns, and new methods for creating a more suitable physical environment temperature to prevent the occurrence of rotational errors.

An implication of staging is on the self-repairing properties of a system. Although we observed the 2D  $3 \times 3$  square being able to self-repair, this was only within the second stage. Further research into features that allow for, and the understanding of the limits to, self-repair between specific stages is required to continue to further develop our approach. For example, although salamanders undergo development through unique stages, they can regrow lost limbs by repeating earlier developmental stages (Wolpert, 1998).

Nevertheless, we envision our staging strategy being applicable to a variety of applications relying on fixed components, such as the design of nano and microscale structures, circuit design, and DNA computing using self-assembly. Moreover, we envision our staging strategy as an approach to improve the ability of artificial evolution for the creation of more complex physical self-assembling systems.

## Conclusions

Staging is an essential part of biological development. In this work we presented a novel approach to staging the self-assembly process using morphological information. This work involved creating two new staged self-assembly analytical tools, the 2DscTAM and the 3DscTAM. Furthermore, this work showed how the interplay between component morphological information (shape and magnetic patterns) can be used to reduce assembly errors and leverage rotational properties by using staging. We presented four proof-of-concept experiments to demonstrate that our staging strategy is a viable method for enabling the self-assembly of more complex morphologies not otherwise possible.

## References

Adleman, L., Cheng, Q., Goel, A., and Huang, M.-D. (2001). Running time and program size for self-assembled squares. In Vitter, J. S., Spirakis, P., and Yannakakis, M., editors, *STOC 2001*, pages 740–748, New York, NY. ACM.

Bhalla, N., Bentley, P. J., and Jacob, C. (2010). Evolving physical self-assembling systems in two-dimensions. In Tempesti, G., Tyrrell, A. M., and Miller, J. F., editors, *Proc. of the*

*Int. Conf. on Evolvable Systems*, pages 381–392, Berlin, Germany. Springer-Verlag.

Cox, D. R. and Snell, E. J. (1989). *Analysis of Binary Data*. Chapman and Hall/CRC, Boca Raton, FL, 2nd edition.

Demaine, E. D., Demaine, M. L., Kekete, S. P., Ishaque, M., Rafalin, E., Schweller, R. T., and Souvaine, D. L. (2008). Staged self-assembly: nanomanufacture of arbitrary shapes with  $o(1)$  glues. In Garzon, M. and Yan, H., editors, *DNA Computing*, Lecture Notes in Computer Science, pages 1–14. Springer Berlin / Heidelberg.

Doty, D. (2009). Randomized self-assembly for exact shapes. In *FOCS '09*, Washington, DC. IEEE Computer Society.

Doursat, R. (2008). Programmable architectures that are complex and self-organized: from morphogenesis to engineering. In Bullock, S., Noble, J., Watson, R., and Bedau, M. A., editors, *Artificial Life XI*, pages 181–188, Cambridge, MA. MIT Press.

He, Y., Ye, T., Su, M., Zhang, C., Ribbe, A. E., Jiang, W., and Mao, C. (2008). Hierarchical self-assembly of dna into symmetric supramolecular polyhedra. *Nature*, 452:198–201.

Ingber, D. E. (1998). The architecture of life. *Scientific American*, 278(1):48–57.

Kao, M.-Y. and Schweller, R. (2006). Reducing tile complexity for self-assembly through temperature programming. In *Proc. of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 571–580, New York, NY. ACM.

Miyashita, S., Nagy, Z., Nelson, B. J., and Pfeifer, R. (2009). The influence of shape on parallel self-assembly. *Entropy*, 11:643–666.

Pelesko, J. A. (2007). *Self Assembly: the Science of Things that Put Themselves Together*. Chapman and Hall/CRC, Boca Raton, FL.

Rieffel, J. and Pollack, J. (2005). Evolutionary fabrication: the emergence of novel assembly methods in artificial ontogenies. In *Proc. of the Genetic and Evolutionary Computation Conference, SEEDS Workshop*, pages 265–272, New York, NY. ACM.

Rothmund, P. W. K. and Winfree, E. (2000). The program-size complexity of self-assembled squares. In Yao, F. and Luks, E., editors, *STOC '01*, pages 459–468, New York, NY. ACM.

Whitesides, G. M. and Gryzbowski, G. (2002). Self-assembly at all scales. *Science*, 295(5564):2418–2421.

Winfree, E. (1998). Simulations of computing by self-assembly. In *DNA Based Computers*, volume IV.

Winfree, E., Liu, F., Wenzier, L., and Seeman, N. (1998). Design and self-assembly of two-dimensional dna crystals. *Nature*, 394(6):539–544.

Wolpert, L. (1998). *The Principles of Development*. Oxford University Press, Oxford, UK.

Wu, H., Thalladi, V. R., Whitesides, S., and Whitesides, G. M. (2002). Using hierarchical self-assembly to form three-dimensional lattices of spheres. *J. Am. Chem. Soc.*, 124:14495–14502.

# Discovering and Maintaining Behaviours Inaccessible to Incremental Genetic Evolution Through Transcription Errors and Cultural Transmission

James M. Borg<sup>1</sup>, Alastair Channon<sup>1</sup> and Charles Day<sup>1</sup>

<sup>1</sup> Research Institute for the Environment, Physical Sciences and Applied Mathematics, Keele University, ST5 5BG, UK  
{j.borg, a.d.channon, c.r.day}@epsam.keele.ac.uk

## Abstract

In this work the question of whether the introduction of both transcription errors and cultural transmission, in the form of learning by imitation, can enable the evolution of behaviours *inaccessible* to incremental genetic evolution alone is assessed. To answer this a neural network model using a hybrid of two different networks was implemented: one capable of demonstrating reactive qualities, the other controlling deliberative goal selecting behaviours. Animats using this model were evolved in an adaptation of the environment proposed by Robinson et al. (2007) to solve increasingly difficult tasks. Simulations were run on populations with and without learning by imitation to assess the relative success of each strategy, leading to the conclusion that populations with learning by imitation can successfully demonstrate the most complex behaviour, which was empirically found to be inaccessible to non-learning populations.

## Introduction

In this paper we present work showing animats in a virtual environment learning behaviours through imitation that are inaccessible to incremental genetic evolution alone. Learning by imitation is often considered to be a mechanism of social information transfer (Cavalli-Sforza and Feldman, 1981; Whiten and van Schaik, 2007), leading to what may be described as social or cultural learning. By combining population learning and individual learning in the same evolutionary system it is possible to make use of both global and local search: global search through the underlying (multi-generational) genetic algorithm and local search through individual (lifetime) learning (Hinton and Nowlan, 1987). It has been demonstrated by Best (1999) that by using cultural learning in place of individual learning on a more challenging version of the Hinton and Nowlan (1987) problem, it is possible to improve the speed at which a population of agents discover an adaptive goal. Cultural learning has the added advantage of allowing individuals to pass on learnt information to other members of the population, and so preserving extra-genetic information for the next generation. Beyond its uses in evolutionary optimisation and search, cultural and social learning is also a well known natural phenomenon with various species using social learning mech-

anisms such as imitation, emulation, teaching and the use of public information to produce adaptive behaviours in dynamic and challenging real world environments (Whiten and van Schaik, 2007; Reader and Biro, 2010).

A number of studies have investigated the effect learning by imitation has on populations of evolving neural networks (Best, 1999; Cangelosi et al., 2006; Acerbi and Parisi, 2006; Acerbi and Nolfi, 2007; Curran and O’Riordan, 2007; Marriott et al., 2010). In much of the literature these imitating neural networks are referred to as agents, with some, as is the case in this work, even taking on the role of animats or autonomous agents in virtual environments (Marriott et al., 2010). It is the aim of this work to investigate whether learning by imitation in a population of neural networks enables behaviours that are deemed to be inaccessible to incremental genetic evolution, to be learned and maintained. In order to test our claims an increasingly complex virtual environment is used in which animats’ behaviours are evaluated. It is expected that without learning these animats will only be able to exhibit a limited set of behaviours, whereas animats learning through imitation should evolve in such a way to allow access to all categories of behaviour.

## Incremental Genetic Evolution

Long-term incremental evolution necessarily uses converged populations, which can be referred to as species (or quasi species). In genetic algorithms (GAs) this is referred to as the Species Adaptation Genetic Algorithm or SAGA approach (Harvey, 2001). The SAGA approach impacts on the way populations evolve: recombination will have a far smaller effect on the motion of the population than in a standard GA, as each species is already genetically similar, leaving mutation as the primary driving force behind evolution. Mutation can be substantially effective in spaces percolated by neutral networks: pathways of level fitness through the fitness landscape. In this case genotypes can vary while still producing similar phenotypes and behaviours. When phenotypes of higher fitness are found the population converges onto them. This incremental approach enables species of animats to discover and converge upon an easily accessible

solution. However, if there is no neutral or incremental path between the corresponding basic behaviour and fitter ones, the population will struggle to move away from these sub-optimal behaviours. Figure 1 depicts a mock example.

One approach to solving the problem of sub-optimal convergence is to increase the rate at which mutation is applied, potentially allowing the population to explore more of the fitness landscape and so discover new fitness peaks. However, there are problems with this approach: as mutation rates increase, the evolutionary search strategy begins to resemble random search, with larger mutation rates making it increasingly difficult for the population to maintain solutions. The point at which mutation becomes so large that favourable structures discovered by evolution are lost more frequently than they are found is known as the error threshold. Ochoa et al. (1999) and others have demonstrated a link between error thresholds and optimal mutation rates in evolutionary algorithms.

### **Discovering and Maintaining Inaccessible Solutions: Transcription Errors and Imitation**

To solve the issue of sub-optimal population convergence without crossing the error threshold, noise is often added to the fitness landscape via the genotype to fitness map. However, where such noise is in the phenotype to fitness section of that map, its ability to aid in the transition between peaks (or more accurately between neutral networks) is limited. By instead incorporating noise into the genotype to phenotype map, as with transcription errors, behaviours inaccessible to incremental genetic evolution may be exhibited reliably by individuals while leaving the genotype untouched. It can be useful to view such noise as a type of unguided individual learning.

In order to maintain successful behaviours in the population, some form of extra-genetic learning needs to take place. The model employed in this work makes use of imitation through interactions between teachers and pupils to facilitate the transmission of learnt behaviours (Cangelosi et al., 2006; Acerbi and Parisi, 2006; Acerbi and Nolfi, 2007; Curran and O’Riordan, 2007). As in Curran and O’Riordan (2007) pupils follow teachers in a mock evaluation on a set of environments. As both teacher and pupil receive the same environmental input the teacher’s output may be used as a target pattern for error backpropagation, reducing the pupil’s output error compared to that of the teacher. By learning in this way pupils are able to imitate the behaviours exhibited by teachers, thus maintaining behaviours in the population that would have been lost in incremental genetic evolution.

### **Neuroevolution of Deliberative Behaviours**

This work uses populations of neural networks embodied in animats. The neural network architecture used here is a hybrid of two different networks: the first controlling the high level deliberative behaviours of the animat, and the sec-

ond controlling the animat’s reactive capabilities (Robinson et al., 2007). By making use of both reactive and deliberative mechanisms, neural architectures of this sort are able to seek long term goals while also reacting to unforeseen events ultimately enabling the evolution of complex problem solving abilities. To demonstrate these problem solving abilities Robinson et al. (2007) developed a complex problem called the ‘river crossing’ or RC task. The RC task required animats to find a single reward-giving *Resource* in a 2D grid-world environment containing a number of obstacles. Alongside *Resource* objects animats could encounter *Water*, *Grass*, *Traps* and *Stones*. Grass objects made up the majority of the environment and were seen as neutral space for the animats to move across; Trap objects were immediately lethal, as were Water objects, which were placed in such a way to resemble an unbroken river cutting the animat’s path to the *Resource*. In order to cross the river animats were required to pick up Stone objects, which could be carried at no cost to the animat, and place them in the same cells as Water thus negating their lethality. Once a continuous bridge of Stones over the river had been built animats could access the *Resource*. To succeed at the RC task animats were required to evolve with no *a priori* knowledge of the world; each new environment was unique and animats had no concept of co-ordinates, making solutions such as ‘move five steps to the right’ impossible, instead animats evolved goals and sub-goals such as ‘go to resource’, ‘avoid traps’ or ‘head to nearest stone’ which then allowed the network to navigate the animat towards these goals. Despite the RC task being reasonably complex, Robinson et al. (2007) demonstrated that it could be solved by initially converged populations of animats using only incremental genetic evolution. To test our hypothesis a more complex version of the RC task has been developed: the RC+ task.

### **The RC+ Task**

An important aspect of the RC task was that individuals were evaluated on increasingly difficult environments. In Robinson et al. (2007), animats were first shown a map with no river blocking their path; then a river with a width of one cell was introduced, followed by a final environment containing a river with a width of two cells. Stone and Trap objects were of a consistent number throughout all tests giving animats equal exposure in each environment. The RC+ task makes the task harder in regard to both river width and exposure to Stone objects. The number of environments an animat is evaluated on is increased from three to five, with environments becoming increasingly difficult to solve due to river width increasing from zero cells to four cells. To add to the difficulty further, the number of Stone objects gradually decreases from twenty in the first environment to zero in the final environment, making each environment more challenging to the point where the final environment cannot be completed by building a bridge. In order to make the final envi-

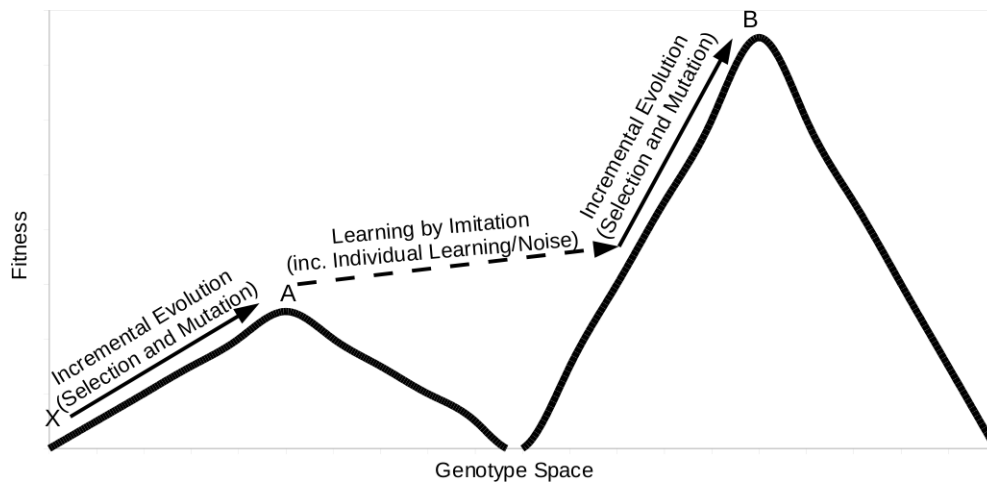


Fig. 1: A species starting from point X on the above mock fitness landscape would achieve peak A by way of the hill climbing strategy adopted by incremental genetic evolution (driven primarily by mutation and selection). Gradient-based learning amongst such a species would ordinarily also be restricted to peak A. The inclusion of both noise in the genotype to phenotype map and learning by imitation can enable the species to jump across areas of lower fitness to higher peaks (inaccessible to hill climbing alone), where incremental genetic evolution and learning can resume hill climbing.

environment solvable two extra objects, *Object A* and *Object B*, are introduced into the environment. *Object A* and *Object B* are rare objects, with only one instance of each found in each environment. Like Stones, *Object A* and *Object B* may be carried at no cost to the animat and placed upon any square or object. If an animat happens to place both *Object A* and *Object B* on a square containing Water (notionally forming a floating raft that carries the animat to the resource), a reward equal to that of the Resource is received and the animat is considered to have successfully solved the environment. In short, an alternate Resource may be constructed out of the three other objects (*Object A*, *Object B* and Water), removing the need to build bridges but still requiring agents to be driven towards the Resource when Water is not present. The RC+ task is impossible to solve with incremental genetic evolution alone. To solve it, animats are required to engage with Water, *Object A* and *Object B* while still avoiding Traps and uncovered Water, and to also be able to reach the Resource in the absence of Water (the simplest sub-solution to evolve). The rarity of both *Object A* and *Object B* adds to the difficulty of the RC+ task as animats must now evolve to be driven to towards *Object A* and *Object B* despite potentially very little exposure during their time in the environment.

### The Model

Animat movement is controlled by a hybrid neural network embodying both reactive and deliberative qualities. This hybrid network may be broken down into two network models: a *shunting network* and a *decision network*, with the decision network passing information on to the shunting network which in turn controls the animat's movement. The

shunting network is not directly exposed to any evolution or learning. The deliberative network on the other hand is exposed to both evolution and learning, enabling the evolution and inheritance of animat behaviour.

### The Shunting Network

Shunting networks are a specialised form of neural network making use of what is known as the shunting model (Yang and Meng, 2000). The inspiration for the shunting model came from Yang and Meng's (2000) desire to develop motion planning systems capable of reacting quickly in real-time environments, thus allowing robotic agents to exhibit robust and collision-free motion planning behaviours. Instead of directly specifying behaviours, the shunting model maps network outputs onto environmental outputs (within an internal map of the environment) which are propagated across the environment to form an activity landscape. This activity landscape is used by the agent to control movement through the environment, by dynamic gradient ascent of the landscape. In their model, Yang and Meng (2000) demonstrated a neural network composed of an  $n$ -dimensional lattice of neurons, with each neuron representing a possible state in the system. By using neurons to represent states in this way it is possible to represent any system which is capable of being fully described by a set of discrete states.

The environment used for the RC and RC+ tasks is a simple 2D grid-world consisting of  $20 \times 20$  cells, with each cell representing a position in co-ordinate space. Each position in the grid-world may be occupied by any number of objects found in the RC+ environment (Resource, Water, Trap, Grass, *Object A* and *Object B*), allowing the system to be fully described by a set of discrete states, thus enabling the

use of the shunting model to direct animat movement across the RC+ environment and ensuring a simple one-to-one relationship between neurons and geographical locations.

In Yang and Meng (2000), two versions of a transition function for specifying inter-neuron dynamics were developed: one which controlled activity saturation in the network and one which did not. Consistent with the findings of Robinson et al. (2007), we found activity saturation not to be a problem exhibited by networks in the RC+ task, enabling the use of the simpler transition function in equation 1.

$$\frac{dx_i}{dt} = -Ax_i + I_i + \sum_{j=1}^k w_{ij} [x_j]^+ \quad (1)$$

Alpha ( $A$ ) represents the passive decay rate, which determines the degree to which each neuron's activity diminishes towards an idle state. The functions  $[x]^+$  is  $\max(0, x)$ . The connection weight (or synapse strength)  $w_{ij}$  between neurons  $i$  and  $j$  is the Euclidean distance between cells  $i$  and  $j$  within the receptive field.  $k$  is the receptive field size and here is set to 4, corresponding to the four cells orthogonally surrounding cell  $i$ . Iota ( $I$ ) is equal to  $E$  in the case of the target, and  $-E$  for an obstacle, where  $E$  is a large integer.

In the case of the RC and RC+ tasks Iota values are limited to 15, -15 and 0, representing the target resource, an obstacle and neutral space respectively. The result of using a transition function with these values are 2D environments with large peaks at the sites of target states, large troughs in cells occupied by obstacles, and large amounts of neutral space through which neuron activity from targets may spread. Using the shunting model to control animat movement allows for goals such as 'head for resource while avoiding traps' or 'place carried stones on water' to be easily achieved.

## The Decision Network

The role of the decision network is to set the Iota values for object states found in the RC and RC+ task. Using the decision network animats can set the desirability of object states in relation to their current environmental inputs, allowing them to manipulate the shunting network's activity landscape and so combine multiple actions such as 'pick up the closest stone' and 'place stone on water' to create complex behaviours.

As in Robinson et al. (2007), the decision network is simply a feed-forward multi-layer perceptron with one hidden layer comprising of four hidden units. The input layer is capable of representing the animat's current state in the environment including whether or not the animat is currently carrying a movable object (Stone, Object A, Object B), with each movable object having a dedicated carrying input. Inputs taken by the input layer are single values of 1 or 0, representing the presence of the object in the same cell as the animat. These input values are fed through to the hidden layer neurons via weighted connections in the range

$[-1,1]$ . At each hidden unit the weighted sum of inputs is passed through a hyperbolic tangent activation function to produce hidden layer outputs. In the RC+ task the output layer is made up of sixty-seven neurons representing the Iota values of all sixty-four possible environmental states (excluding Grass objects whose Iota values are always set to 0 and therefore do not need be represented in the decision network) and a pick-up/put-down output for each non-static object (Stone, Object A, Object B). At each output neuron the sum of all weighted connections is passed through a hyperbolic tangent activation function with fixed thresholds: neurons outputting within the range  $[-0.3:0.3]$  are set to output 0, while all outputs over 0.3 resolve to 1 and all outputs below -0.3 resolve to -1.

For outputs representing the pick-up/put-down actions output values of -1 cause the animat to put down the specified object they are carrying, values of +1 causing animats to pick up the movable objects they are currently sharing a cell with providing the animat is not already carrying an object of that type. For all other outputs, resolved output values set the Iota values to be used in the shunting network. So if an output neuron has a negative output, all objects of that class found in the environment at that point in time will have their activations set to -15; for positive outputs to +15. Any object resulting in an Iota value of 0 will remain neutral, causing their activation values in the shunting network to be solely based on the propagated activations of other objects. The resulting environment will contain a number of peaks of high activity and troughs of low activity, gradually propagating activity through neighbouring neutral cells.

Figure 2 shows two of the five potential environments an animat may observe in the RC+ task, and the corresponding activity landscapes given certain outputs from the decision network. The first environment represents the initial challenge an animat must complete, where only traps stand in the way of a resource. As can be seen by this environment's activity landscape, the Iota value associated with the resource has been set to be positive resulting activity propagating from the resource over the surrounding neutral space. The second environment represents the second challenge, to cross a river before having access to the resource. In this environment's case, activation propagation from the resource has been impeded by the decision network outputting negative Iota value for Water objects. Negative activity repels animats from objects with negative Iota values; however positive activation can be seen coming from the Object B object, providing a hill-climbing route for the animat to take in activity space.

## Evolution of the Decision Network

To evolve the decision network a steady-state genetic algorithm was used. At each iteration two animats were selected from the surviving population to be evaluated in tournament selection, with the worst performing animat being

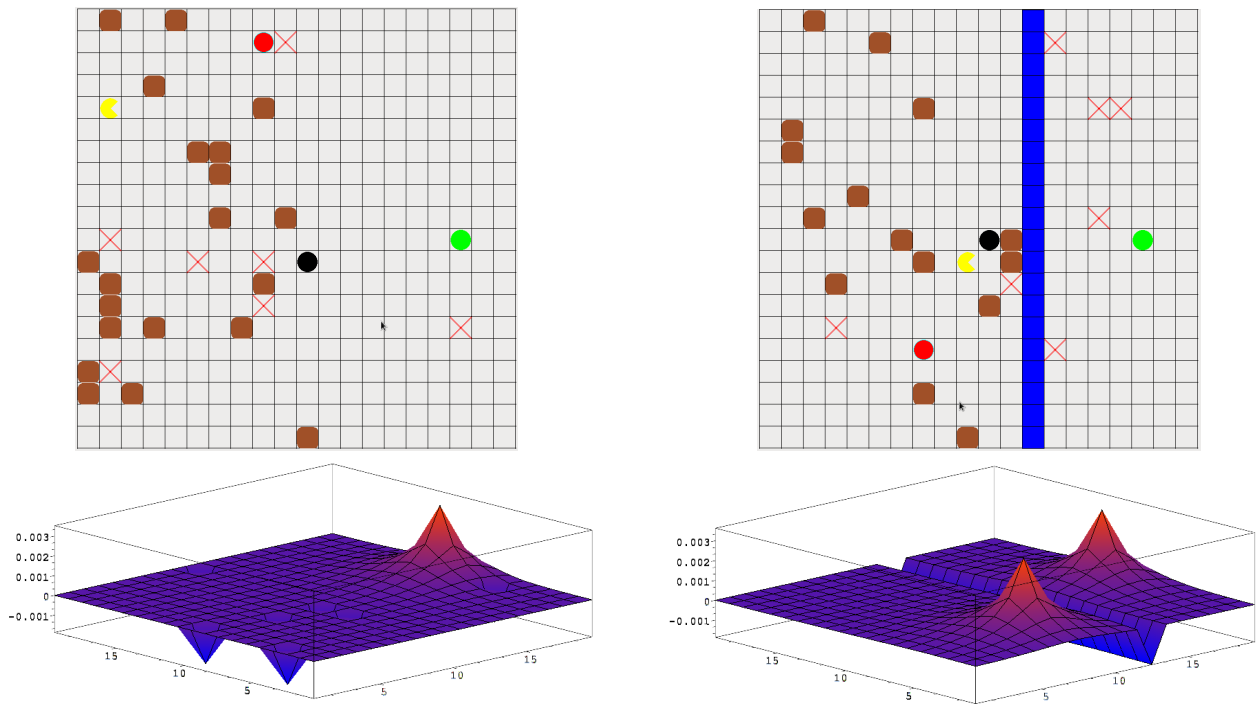


Fig. 2: Two environments with their activity landscapes (given certain outputs from the decision network - see main text). Animat=yellow, Stones=brown, Resource=green, Object A=black, Object B=red, Traps=crosses, Water=blue.

replaced by the progeny of the better performer. The competing animats are evaluated in five increasingly difficult environments. If during evaluation an animat fails to complete an environment, the evaluation is terminated. Fitness is set to be the number of environments successfully completed by an animat during evaluation.

An animat's genotype consists of a set of floating point values each in the range  $[-1,1]$ , which are transcribed into the connection weights in the animat's decision network. The genotype and the decision network are stored separately, so any learning that may take place during an animat's lifetime will only affect the decision network: no changes are made to its genotype after an animat is initially created. New animats are the offspring of two other animats from the current population: one tournament winning animat and one randomly selected animat. The child's genotype is created first through recombination of the parents' genotypes; for this operation single-point crossover is used with the point of crossover being a randomly selected point in either parent's genotype. Each loci in an animat's genotype represents exactly the same connection weight as in any other animat's genotype, with all genotypes being of length  $L = 308$ . Mutation follows recombination; each point has a probability  $P_{\text{mut}} = 1/L$  of having a random value from  $N(0,0.4)$  added to it, with the resulting values being bounded within the range  $[-1,1]$ . Once the genotype has been constructed it is written to the new animat's decision network; this process is

referred to as transcription. During transcription two randomly selected connection weights are overwritten with a new random value selected from a discrete uniform distribution  $U(-1,1)$ . The weights now present in the decision network dictate the animat's future behaviours within each environment.

### Learning in the Decision Network

Following reproduction new animats are afforded the opportunity to learn from a teacher via error-backpropagation. This method of teacher-pupil backpropagation has been previously employed by Curran and O'Riordan (2007). However, the teacher-pupil scenario used in this work differs in a number of ways. In the learning model used by Curran and O'Riordan (2007), teachers were selected from the population based upon their fitness and then assigned  $n$  pupils to teach. We contend that in nature absolute fitness is very difficult to assess. To resolve this issue, the current tournament-winning parent is assigned the role of teacher, with the parent's most recent progeny assigned the role of pupil.

There are also differences in the way error-backpropagation is used to teach pupils in this model compared to that of Curran and O'Riordan (2007). As with our model, Curran and O'Riordan (2007) allowed pupils to hitchhike on the back of the teacher during a mock evaluation, with inputs shared between teacher and pupil and using the teacher's output pattern as a target pattern

for the pupil to learn. The learning method employed by Curran and O’Riordan (2007) permitted pupils to learn from the target pattern until the error between child and parent outputs were minimised to a satisfactory level. In our model pupils are only presented with the current teacher’s output once every simulation time step (immediately after the teacher’s decision network’s inputs, activations and outputs are updated). If a teacher happens to move through the environment in such a way that both inputs and outputs remain the same, the child will be presented with many opportunities to learn a given target input-output pattern. However, if the teacher moves around the environment via many different input combinations, the student will have the opportunity of potentially witnessing many different target outputs but at the cost of having very little time to minimise error. Imitating in this manner enables the population to retain favourable behaviours not coded for genetically, whilst not undermining the incremental genetic evolutionary process.

## Experimentation

At each iteration of the model two individuals are taken from the population to be evaluated on a series of five environments/maps. All maps have seven Trap objects placed randomly on the map, one reward-giving Resource, one Object A, one Object B, and  $20 - (5 \times \text{riverwidth})$  Stone objects. River width varies from an initial width of zero, increasing by one cell per map. During evaluation individuals must successfully reach the Resource or place Object A and Object B onto a cell containing Water; any animat failing to do so within 100 steps or dying by means of a Trap or uncovered Water is not permitted to attempt the next environment.

Fitness in the model is determined to be the number of maps successfully completed in the current tournament iteration, with individual fitness being set to zero before each evaluation. The individual achieving the highest fitness is allowed to reproduce, with the weaker individual being replaced by the progeny of the tournament winner and a randomly selected animat. This steady-state approach maintains the population at a size of 100 individuals.

After reproduction the child is allowed to learn via error-backpropagation from its tournament winning parent. The child follows its parent in a mock evaluation, with the child’s inputs being set to those of the parent. Learning takes place for as long as the parent is being evaluated. Once the parent either fails to complete a map or completes all five environments, learning is terminated. At each step through the evaluation the child attempts, via error-backpropagation with a learning rate of  $\delta = 1$ , to learn to imitate the parent’s output for the current inputs.

Three strategies are used in this model: two without learning and one with learning. Populations of animats with no access to learning fall into two categories. The first, known as *Non-Learners(1)*, having a mutation rate and transcrip-

tion error equal to that use by learning populations. As populations of *Non-Learners(1)* have no way of assimilating transcription errors back into the genotype it may be seen as giving learning populations, known as *Learners*, an unfair advantage. With this in mind a second of category of non learners, known as *Non-Learners(2)*, are also evaluated. *Non-Learners(2)* do not have transcription errors, and instead have a mutation rate equal to that of the original mutation rate plus two transcription errors:  $P_{mut2} = 3/L$ .

To test the ability of each strategy to exhibit the behaviour necessary to complete the most difficult map, fifteen populations of each learning strategy were simulated. Each simulation lasted a maximum of 5,000,000 tournaments. In each simulation the best individual’s fitness and the mean population fitness were recorded at intervals of 500 tournaments. The maximum fitness an individual could achieve was five, which directly relates to the successful completion of all five evaluation environments, the fifth environment being impossible to complete by bridge building and so requiring the combination of Object A and Object B on Water. For a population to be considered as adequately completing the fifth map, a fitness of five must have been recorded by the fittest individual at ten recorded tournaments with at least five of these tournaments being unbroken by a sub-optimal result. This ensures that the complex behaviour tested for is not only found but also maintained by the population.

## Results

Table 1 shows results from the fifteen populations of animats using the *Non-Learners(1)* strategy: the mean, best and worst number of tournaments required to solve each map, across the fifteen populations (runs), and the proportion of populations that were successful in solving each map. Of the *Non-Learners(1)* populations over 90% were able to complete maps 1 to 4 but no population was able to demonstrate a successful solution to map 5. Populations of animats using the *Non-Learners(2)* strategy also demonstrated a high level of proficiency when completing maps where the bridge building solution is effective, though with a lower proportion of populations able to complete map 4 (see table 2). This may be due to the higher mutations rate used in the *Non-Learners(2)* strategy causing the destruction of potentially beneficial behaviours before they can proliferate through the population. To complete map 4 animats had to be stricter (more consistent) in their use of Stone objects. Despite this behaviour being reachable using incremental genetic evolution it is within a small area of weight-space, causing it to be potentially lost with higher mutation rates. Neither non-learning strategy was able to discover the precise behaviour necessary to complete map 5, so failures recorded in tables 1 and 2 were not due to a sufficient behaviour being discovered but not maintained: the map 5 solution was simply never found, empirically demonstrating the inaccessibility of map 5 to incremental genetic evolution alone.



Map	Mean	Best	Worst	Stdev	Success
1	1200	500	3500	996	100%
2	502571	11000	2152500	738090	100%
3	1568000	34000	4429500	1501336	93%
4	1613786	58000	4432500	1506065	93%
5	N/A	N/A	N/A	N/A	0%

Tab. 1: Non-Learners(1): Mean, best, worst number of tournaments required to solve each map.

Map	Mean	Best	Worst	Stdev	Success
1	1400	500	3000	784	100%
2	81692	4500	252500	96805	100%
3	1801286	12500	4987000	1502754	93%
4	2193385	41500	4466500	1497156	87%
5	N/A	N/A	N/A	N/A	0%

Tab. 2: Non-Learners(2): Mean, best, worst number of tournaments required to solve each map.

Table 3 shows results from animats using the Learners' strategy. Unlike non-learning strategies, Learners are able to complete map 5 and thus exhibit the complex behaviour tested for in this work a third of the time, proving the hypothesis that learning by imitation is capable of enabling populations of animats to discover behaviours found to be inaccessible to incremental genetic evolution alone. However, Learners are seemingly less likely to discover and maintain solutions to maps 3 and 4 than non-learning animats.

Figure 3 charts the mean fitness of the best performing population from each learning strategy. From this graph it can be observed that Learners bypassed the sub-optimal bridge building solution once the population had (for some time) been evaluated on maps with rivers. The incremental nature of the evolution in this model causes the majority of the population to rapidly converge on the optimal solution once it has been discovered. Without learning, this optimal behaviour cannot be found. In this model incremental genetic evolution leads to convergence on sub-optimal solutions in non-learning populations, making it impossible for the discovery of the optimal behaviour. By combining learning by imitation and incremental genetic evolution in a

Map	Mean	Best	Worst	Stdev	Success
1	1533	500	5000	1302	100%
2	512333	9500	2026000	616376	100%
3	2484455	5600	4340500	1395760	73%
4	2458800	88500	4211500	1861794	33%
5	1843200	83500	3851000	1631808	33%

Tab. 3: Learners: Mean, best, worst number of tournaments required to solve each map.

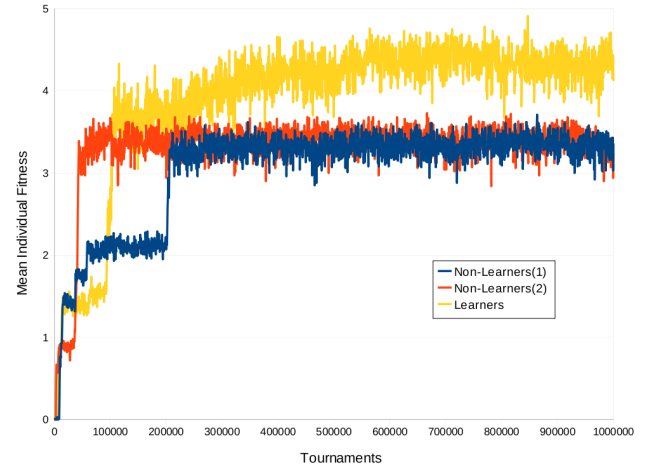


Fig. 3: Graph showing the mean fitness in the best performing populations for each learning strategy. Populations learning by imitation demonstrated the ability to converge on more complex behaviours, thus achieving a higher fitness. Neither non-learning strategy is capable of producing the more complex behaviour.

model such as the one presented here, it is possible to not only discover complex behaviours inaccessible to incremental evolution alone, but also to have rapid convergence to a population exhibiting and maintaining that behaviour, thus creating a behavioural tradition or culture (Whiten and van Schaik, 2007). The results found here are broadly consistent with those of Acerbi and Nolfi (2007), who found that the combination of individual and social learning in artificial embodied agents not only allowed for the development of difficult and costly behaviours, but also provided an adaptive advantage over individual learning alone and lead to cumulative cultural evolution.

## Conclusions and Future Work

If a learnt behaviour is exhibited and maintained throughout a population for a number of generations it may tentatively be called a tradition or even a culture. According to Whiten and van Schaik (2007) traditions are “consistent habits” that make use of social information transfer. In the model demonstrated here learning by imitation enables social information transfer with behaviours being maintained by converged populations or species giving rise to traditions. The limited set of behaviours observed in this population do not however constitute the category of culture, which is reserved for the maintenance of multiple behaviours by a species. The incremental nature of the model causes sub-optimal behaviours to be phased out of the population. Were greater environmental diversity to be used, it may be possible to evolve a culture rather than a tradition.

The hypothesis presented here was that the introduction of both transcription errors and cultural transmission in the



form of learning by imitation are sufficient to discover and maintain the most complex behaviour possible in the model, while incremental genetic evolution alone is not. The results prove our hypothesis by demonstrating that without learning by imitation the solution to the final environment is never found but with imitative learning all behaviours can be discovered, exhibited and maintained.

One drawback to the model used in this work is the limited set of behaviours available to animats. By using a larger environment with a greater variety of potential states available to the animats and evolving the size and structure of the decision network, it may be possible to demonstrate the evolution of multiple behaviours leading to the emergence of a culture. To investigate more complex behavioural development and the role of imitative learning in the evolution of traditions and cultures, it would be beneficial to implement larger and more dynamic environments and allow for greater evolution in the decision network. A secondary drawback was the simple vertical social transmission mechanism used. The inclusion of intra-generational or oblique cultural transmission has been shown to be both sufficient (Cavalli-Sforza and Feldman, 1981) and beneficial (Acerbi and Parisi, 2006) for the evolution of complex and robust cultural behaviours. Further investigation and application of oblique transmission within models such as that presented here would further benefit our understanding of and ability to achieve the evolution and maintenance of complex cultural traits.

## References

- Acerbi, A. and Nolfi, S. (2007). Social learning and cultural evolution in embodied and situated agents. In *Proceedings of the IEEE Symposium on Artificial Life, ALIFE'07*, pages 333–340. IEEE.
- Acerbi, A. and Parisi, D. (2006). Cultural transmission between and within generations. *Journal of Artificial Societies and Social Simulation*, 9(1).
- Best, M. (1999). How culture can guide evolution: An inquiry into gene/meme enhancement and opposition. *Adaptive Behavior*, 7(3–4):289–306.
- Cangelosi, A., Hourdakis, E., and Tikhanoff, V. (2006). Language acquisition and symbol grounding transfer with neural networks and cognitive robots. In *Proceedings of the 2006 International Joint Conference on Neural Networks, IJCNN'06*, pages 2885–2891.
- Cavalli-Sforza, L. L. and Feldman, M. (1981). *Cultural Transmission and Evolution: Quantitative Approach*. Princeton University Press.
- Curran, D. and O’Riordan, C. (2007). The effects of cultural learning in populations of neural networks. *Artificial Life*, 13(1):45–67.
- Harvey, I. (2001). Artificial evolution: A continuing saga. In *Proceedings of the International Symposium on Evolutionary Robotics From Intelligent Robotics to Artificial Life*, ER ’01, pages 94–109. Springer-Verlag.
- Hinton, G. and Nowlan, S. (1987). How learning guides evolution. *Complex Systems*, 1:495–502.
- Marriott, C., Parker, J., and Denzinger, J. (2010). Imitation as a mechanism of cultural transmission. *Artificial Life*, 16(1):21–37.
- Ochoa, G., Harvey, I., and Buxton, H. (1999). Error thresholds and their relation to optimal mutation rates. In *Proceedings of the 5th European Conference on Advances in Artificial Life, ECAL ’99*, pages 54–63. Springer-Verlag.
- Reader, S. and Biro, D. (2010). Experimental identification of social learning in wild animals. *Learning and Behavior*, 38(3):265–283.
- Robinson, E., Ellis, T., and Channon, A. (2007). Neuroevolution of agents capable of reactive and deliberative behaviours in novel and dynamic environments. In *Proceedings of the 9th European conference on Advances in artificial life, ECAL’07*, pages 345–354. Springer-Verlag.
- Whiten, A. and van Schaik, C. (2007). The evolution of animal ‘cultures’ and social intelligence. *Phil. Trans. R. Soc. B*, 362(1480):603–620.
- Yang, S. and Meng, M. (2000). An efficient neural network approach to dynamic robot motion planning. *Neural Networks*, 13(2):143–148.

# Modeling and Simulating Crowdsourcing as a Complex Biological System: Human Crowds Manifesting Collective Intelligence on the Internet

Thierry Buecheler<sup>1</sup>, Rocky Lonigro<sup>1</sup>, Rudolf M. Fuchslin<sup>2</sup> and Rolf Pfeifer<sup>1</sup>

<sup>1</sup>Artificial Intelligence Laboratory, Department of Informatics, University of Zurich

<sup>2</sup>Center for Applied Mathematics and Physics, School of Engineering, Zurich University of Applied Sciences  
buecheler@ifi.uzh.ch

## Abstract

Crowdsourcing, a real-life instance of human collective intelligence, is a phenomenon that changes the way organizations use the Internet to collect ideas, solve complex cognitive problems, and build high-quality repositories (e.g., Wikipedia) by self-organizing agents around data and knowledge. Many recent studies have highlighted the factors and the small sets of parameters that play a role when a large crowd interacts with an organization. However, no comprehensive simulation has yet been developed to incorporate all these parameters, investigate Artificial Life phenomena such as emergence and self-organization and potentially generate predictive power. Based on a presentation at ALIFE XII, this paper describes the development of a simulator for human crowds performing collective problem solving in a Crowdsourcing scenario. It introduces the mechanics of a multi-agent system (MAS) by building on insights from empirical science in several disciplines. The simulator allows running sensitivity analyses of multiple parameters as well as simulation of intractable interactions of complex networks of irrational agents. In addition, the paper provides a review of Crowdsourcing and human collective intelligence literature structured from an Alife point-of-view.

## Introduction

Many researchers in the Artificial Life community are researching self-organizing, decentralized systems (e.g., large groups of ants or vertebrates such as bison [*collective animal intelligence*]) that show, in their interactions, a high degree of (self-)stability and flexibility. Social scientists are transferring these insights to social networks and other interactions between humans (for examples, see Krause et al. 2010).

A Crowdsourcing scenario provides an excellent setting for investigating *human collective intelligence*, generated through networks of interactions among individuals and between individuals and the environment. “Crowdsourcing” (Howe 2008), an instance of collective intelligence (Buecheler et al. 2010, Robu et al. 2009) emerging from de-centralized actions of a community of users, is a phenomenon currently occurring all over the world, strongly benefiting from new technologies and the development of Web 2.0: In essence, a “seeking” entity (e.g., a company or university) seeks the support of an *apriori* unknown and potentially very large group of intelligent agents (i.e., humans) by posting its unsolved problems on the internet. A simple and famous example is the

Wiki Foundation (seeker) using internet users (crowd of solvers) to produce the world’s largest encyclopedia (Wikipedia) with high-quality results (Giles 2005). This can either happen directly, as with Wikipedia, or through “information brokers”, such as Innocentive.com or NineSigma.com that connect seekers and solvers through a platform. Small start-up companies as well as large established institutions (such as Fortune 500 companies and universities or other large organizations like NASA) are currently using Crowdsourcing for a variety of purposes. (Kittur et al. 2008) describe how user input can substantially improve the interaction design and how input after development can provide important feedback for continued improvement based on investigations on Amazon’s “Mechanical Turk”. Some examples where collective human intelligence is more useful than mere computational power by using “games with a purpose” are given in (von Ahn 2006).

The underlying complex dynamics are being intensively investigated by researchers from several disciplines, sometimes using different names like “Open Innovation” (Chesbrough 2003) or “Swarm Intelligence” (Dorigo and Stützle 2004, Krause et al. 2010) for slight variations of the phenomenon of interest. Recent studies investigate single parameters or specific settings of Crowdsourcing (e.g., Sieg et al. 2011, Leimeister et al. 2009, Alonso et al. 2008). However, so far there has been no comprehensive simulation of the complex interactions between agents involved in this scenario.

Phenomena relevant in an Artificial Life context such as self-organization, stigmergy and especially emergence, are very relevant when trying to understand complex dynamics between humans in real-life organizational scenarios (Bandtke 2007). The emerging phenomena in organizations are created through interpersonal, analytically irreducible factors such as spontaneity, informal structures and interactions, ad hoc processes and groups as well as informal conventions such as norms and similar social patterns. This paper describes the underlying dynamics of a complex Crowdsourcing system and an implementation in our simulator, based on a framework for multi-agent systems (MAS). The simulator uses parameters based on empirical studies and integrates a set of essential factors and rules of interactions between members of the “crowd” and the seeking entities. Hence, this paper also provides a structured literature review of Crowdsourcing parameters. Due to its modular set-up, the simulator allows for the addition of more factors, once understood by scholars, to

increase the accuracy of simulation runs and, potentially, its predictive power. The large set of rules and unpredictable outcomes, such as negotiation results between agents (modeled in the MAS), allow the observation and statistical evaluation of emergent phenomena.

In what follows we review the state of the art in Crowdsourcing and Open Innovation research and relevant multi-agent simulation topics, then explain which parameters we are modeling and introduce the simulator before discussing first insights, use cases, and potential next steps.

## State of the Art

We discuss three sections of prior literature: First, the application of swarm behavior and problem solving insights from biology to Crowdsourcing, then relevant insights from management and organization science, often published in the context of “Open Innovation”. In the third part, we look at theoretical and empirical evidence from other contexts that are also important for the creation of this simulator.

### Crowdsourcing, Communities and Group Behavior

(Krause et al. 2010) describes the advantages and challenges of transferring insights from biological studies to human social interactions. Similar to swarms, flocks, and herds, humans follow certain local rules of interaction in large groups. In a Crowdsourcing context, these local rules are evolving over time. Initially, chaotic behavior converges into social patterns and the crowd members use their local knowledge (similar to birds in a flock or fish in a school) to interact with other agents and contribute to Crowdsourcing. In most cases, the crowd self-organizes without a central body of control. Self-organization, as defined by (Camazine et al. 2003), states that “the rules specifying interactions among the system’s components are executed using only local information, without reference to the global pattern”, emerging from lower-level components of the system.

The crowd is especially good at solving coordination or cooperation problems (Surowiecki 2004). (Schelling 1960) investigated the reasons for this and found a possible explanation in focal points (“Schelling points”), towards which human expectations converge, leading to an eventual convergence of actions, comparable to John Dewey’s “cooperative intelligence”. They usually don’t act for the good of the whole crowd, but act according to what’s best for themselves (see Surowiecki 2004). This includes behavior that is judged highly irrational or short-sighted from an outside perspective (see e.g., Simon 1996). Nevertheless, humans can coordinate their actions and achieve complex goals that would not be achievable by individuals (like writing a high-quality encyclopedia or finding a relevant piece of information from billions of web-sites<sup>1</sup>).

From a collective intelligence point of view, cognitive problems (as often appear in a Crowdsourcing context) are even harder to solve than coordination or cooperation problems, because they are often very difficult to centrally organize for a group solution. The solution approach to such

problems is usually emergent and contains almost no formal structuring (see, e.g., *ultimatum* or *common good* games).

### Organizations and Open Innovation

We will use an organizational (more precisely, Open Innovation) context for the basis of the simulation to define more clearly the environment within which our agents are interacting. In this context, Crowdsourcing is often seen as innovation-seeking: Organizations create, acquire, and integrate diverse knowledge and skills required to develop complex innovative technologies. Since knowledge is available from the outside (see, e.g., Chesbrough 2003) the organizations may benefit from leveraging external knowledge. Crowdsourcing is an increasingly popular approach for doing that. Not only are seekers and solvers involved, but also a wide variety of other, intermediate organizations. The acquired capabilities are combined and re-combined without centralized, detailed managerial guidance – again showing a high degree of self-organization. Joel West, one of the first researchers to address “Open Innovation”, defined it as “using the market rather than internal hierarchies to source and commercialize innovations.”

### Other Important Insights

Two important underlying concepts in Crowdsourcing are *private information* and *tacit knowledge*, both emphasizing the “stickiness” of information (information used in technical problem solving is often costly to acquire, transfer, and use in a new location, see von Hippel 1994). An important prerequisite for the success of Crowdsourcing is to maintain the *diversity* of the crowd members’ knowledge and skills throughout the process and to avoid *groupthink*.

**Private information.** (von Hayek 1945) observed that humans possess a special type of local information that is hard to aggregate. Due to such “private information”, nearly every individual “has some advantage over all others because he possesses unique information of which beneficial use might be made.” Crowdsourcing brings together and motivates individuals to collaborate and produce innovation or solve problems based on this private information.

**Tacit knowledge.** Michael Polanyi coined the term in the 1950s using “riding a bicycle” as an example of something humans are able to do “without quite knowing how”. In a Crowdsourcing context, three phenomena can be observed: Communication between different experts may lead to new developments, which implies that each expert’s knowledge is not “tacit”. What Crowdsourcing does is establish new correlations between pieces of knowledge acquired by individuals. A second phenomenon is knowledge that is not “owned” by an individual, or *collective tacit knowledge*. It is neither clear how it can be described nor how it is acquired as a “good of the group”. An example of such collective tacit knowledge under constant change is human natural language. Crowdsourcing not only generates, but potentially also maintains collective tacit knowledge. The third phenomenon derives from the observation that an individual may have knowledge but might not be able to communicate it in a formalized way (“riding a bicycle”). Several studies suggest

<sup>1</sup> Google’s Pagerank algorithm crowdsources a great deal of collective human intelligence to rate the importance of pages by linking to them.

that especially such tacit knowledge and knowledge of technique are best conveyed through collaboration (Lee and Bozeman 2005), as happens in Crowdsourcing. In summary, Crowdsourcing helps locate and productively use these different types of tacit knowledge that cannot be found by the most sophisticated search engines due to its “tacitness”.

The combination of private information and tacit knowledge explains why “irrational” individuals can produce rational outputs: “Rationality” requires some basic assumptions plus logic. In order to apply logic, the assumptions need to be stated in one or the other form of a proposition. Tacit knowledge can be rational and logical, but cannot be stated or codified. Therefore, individual behavior may appear irrational to outsiders. Due to the collective intelligence unearthed during a Crowdsourcing interaction, combining private information (tacit or not) to a solution for a complex problem, the seemingly irrational becomes rational and productive. This implies, however, that optimal group outcomes are hard to achieve because the barrier of perceived individual irrationality needs to be overcome.

#### Expertise, diversity, independence and groupthink.

(Page 2008) found evidence for the advantage of diversity in groups performing complex tasks by running agent-based simulations. His surprising insight was that groups with diverse agents almost always performed better than groups consisting only of expert agents. Herbert Simon (1996) used the term “docile” for individuals who tend to accept information and advice from the social groups to which they belong. He theorized that these individuals have great advantage in fitness over those who are not docile. And it is not easy to stay independent of a social environment since learning is a social process. One can therefore say that although the members of a crowd should show a certain level of docility by e.g., building on previous solutions (if public), the group as a whole should maintain a high diversity of skills and private information. On the other hand, the crowd should avoid “groupthink”. Irving Janis defined the term in the 1970s as follows, based on William H. Whyte’s original 1952 definition: “A mode of thinking that people engage in when they are deeply involved in a cohesive in-group, when the members’ strivings for unanimity override their motivation to realistically appraise alternative courses of action.” On a closed Crowdsourcing platform, where solvers cannot see other solvers’ solutions (which is often the case when monetary premiums are involved) the likelihood of groupthink is clearly much smaller. (Surowiecki 2004) identifies stock bubbles and crashes as famous examples where all of the factors that make crowds smart (independence, diversity, and personal opinion) disappear.

Synthesizing the above sections, once a problem has been formalized and established methodologies exist for its solution (and it is well understood what a solution is), a group of experts may obtain the “correct” solution. However, as long as these methodologies do not exist, no “objective” formulation of the problem has been found and/or evaluation criteria for solutions are not completely formalized, diversity may beat expertise (see discussion and further references in Buecheler et al. 2010). Restricting the group of problem solvers to experts overlooks the fact that this happens at the price of groupthink (acquiring expertise usually includes a certain

level of docility). Maintaining diversity therefore inhibits the very occurrence of groupthink.

## Crowdsourcing in a Nutshell

This section briefly describes the Crowdsourcing process used for our simulator and introduces nomenclature in *italics*. (Muhdi et al. 2010) gives a more detailed description of the activities involved.

- 1. Deliberation:** A *seeker* (an organization<sup>2</sup> or individual) decides to use external sources to generate ideas or solve a specific problem.
- 2. Preparation:** The seeker often chooses an intermediary (information broker), i.e. a website that brings together the seeker and the *crowd* (group of *solvers*) and usually enters into a contract with this information broker.
- 3. Execution:** The seeker posts a *problem* on the Internet. The solvers self-organize and self-select which of all posted problems they would like to work on. The seeker might or might not interact with the solvers during this phase.
- 4. Assessment:** After the execution phase (or perhaps in parallel), the submitted ideas are clustered, rated, and the best idea is rewarded. In our simulator, one to five ideas (the *winning solutions*) receive the *prize premium* (if any).
- 5. Post-processing:** The collected ideas are incorporated into the seeking organization and “side effects” (e.g., creative spillovers useful elsewhere) are managed (if any).

## Complexity and Focus Trade-off in MAS

This simulator attempts to optimize the balance between complexity and focus.

Many simulators for understanding social behavior are based on mathematical models using partial differential equations (PDEs). Multi-agent system simulations have been shown to have certain advantages over these models when simulating large groups of agents. This is also true for a Crowdsourcing model: PDEs cannot handle diverse populations efficiently. Diversity requires particle or agent based models for two reasons: PDEs are increasingly difficult to solve if the number of variables increases and they cannot handle discretization (you either have at least one individual knowing about XY in a group, or this knowledge is not there at all. There is never half of an expert) or stochastic fluctuations as appear in Crowdsourcing. Our simulator is built on modules that incorporate empirically shown parameters in a Crowdsourcing scenario. Thus, it intrinsically simplifies and omits certain properties. However, too much simplification can make a model unnecessarily unrealistic and uninteresting. The simulator includes an adequate level of complexity by incorporating several parameters, while focusing on the parameters that give the “simplest explanation” for the emerging phenomena.

<sup>2</sup> We use “seeker” for any kind of individual or organization (including companies and universities). Science is, in many cases, trying to find inventions or seek evidence for observed/hypothesized phenomena. The processes and paradigms, however, are similar to a large extent.

## Parameters Used for the Simulator

In this section, we show parameters allocated to “seekers”, to the “problem” and finally to “solvers”. In addition, we list some important global parameters used. See Figure 1 for an overview of the most important parameters and their interdependencies. For a detailed description of the parameter representation and spaces please contact the corresponding author.

### Seeker Parameters

The seeker parameters model environmental and internal variables for the seeking organization (e.g., a company or university).

**Degree of revealing and Intellectual Property (IP) regime.** Depending on the strictness of the IP (or “appropriability”) regime (Teece 1986) organizations, especially firms, adopt different formal and informal methods (patents, trademarks, copyright, time-to-market, trade secrets) to adjust their degree of openness. This influences the potential success of the problem in Crowdsourcing: the trade-off between openness to provide maximum information to the solvers while protecting own intellectual property needs to be found. (Henkel 2006; von Hippel and von Krogh 2003) show that openness is not automatically a disadvantage. IBM (according to Kazman and Chen 2009 the most patent-productive company in the world) began making more money from crowdsourced services than from all its patent-protected intellectual property (Benkler 2006). (Dahlander and Gann 2010) has further elaborated on this trade-off in an excellent literature review on Open Innovation while (Lakhani et al. 2006) has shown the importance of openness in a Crowdsourcing context. The difficulties of protecting IP in Crowdsourcing have not yet been resolved.

University researchers tend to be more open (see, e.g., David 2003) but since the introduction of the Bayh-Dole Act in the US in 1980 and similar legislation in other countries, there has been a trend towards more “closed” research. In this context, (Heller and Eisenberg 1998) popularized and discussed the phrase “tragedy of the anticommons”.

**Level of NIH.** (Katz and Allen 2007) described the effect of the “Not-Invented-Here” (NIH) syndrome, the “tendency of a project group of stable composition to believe it possesses a monopoly of knowledge of its field, which leads it to reject new ideas from outsiders to the likely detriment of its performance”, on R&D project groups. This syndrome is clearly relevant in a Crowdsourcing context and it has been shown that a lower level of NIH supports successful internal and external solution development (Brown and Eisenhardt 1995).

**Absorptive capacity.** The term defined in (Cohen and Levinthal 1990) refers to the ability of a seeker to recognize the value of new, external information, assimilate it, and apply it to commercial ends. This parameter is assigned to the seeking agents and will rise over time, when the seeker climbs the Crowdsourcing learning curve and accumulates prior related knowledge (see also Brown and Eisenhardt 1995 for a

discussion). The basic assumption for both NIH and absorptive capacity is that the individuals at the seekers’ interfaces and the crowd co-evolve by exogenous influences and endogenous self-organization (Mitleton-Kelly 2003).

**Historical success rate.** This parameter is used to modify the “Level of NIH” and “Absorptive Capacity” parameters over time. In essence, it shows how far the seeker is on the Crowdsourcing learning curve.

**Crowdsourcing success.** This parameter can only be set as a dependent/output variable. It estimates the success of the product or patent based on the solution/idea gained in the Crowdsourcing process (not incorporating side effects). (Howe 2008) found that an InnoCentive.com company’s average earnings from a successful solution are twenty times the fee paid to a solver. The parameter is influenced by the solution quality, the level of NIH and the absorptive capacity of the seeker. A side remark: The measurement of Crowdsourcing success varies widely between different types of organizations or businesses. Examples are business measures related to finances, employees’ motivation, new product revenue, spending in R&D, number of patents, time to market, and combinations thereof. The build-up of absorptive capacity could in fact already be a goal and measure of success for an organization.

### Problem Parameters

**Problem value – intrinsic.** Several Crowdsourcing interactions do not only target crowds looking for additional income, but also solvers working for intrinsic motivation. Open source developers, for example, show very different motivating factors (see intrinsic motivation factors, below).

**Problem value – monetary.** Most Crowdsourcing platforms assign a prize premium to a problem. Although on many platforms there is no limit, the seeker may divide the premium from zero up to five “winning” solutions in our simulator, in order to constrain simulation complexity (typical is 0 to 10).

**Problem field.** Every posted problem is assigned a primary and (optional) secondary field (or scientific discipline). Examples are “molecular biology” or “organization science”. For our simulator, we used the fields selectable at Innocentive.com (currently the largest Crowdsourcing platform). Solvers do not only work successfully on problems from their respective fields: (Lakhani et al. 2006) found the odds of a solver’s success increased in fields in which they had no formal expertise, confirming a network theory insight from (Granovetter 1973): The most efficient networks are those that link to the broadest range of information,

knowledge, and experience. (Howe 2008) summarizes another important insight from Lakhani’s paper:

A full 75 percent of successful solvers already knew the solution to the problem. The solutions to the problems in the study – many of which [...] had stumped the best corporate scientists in the world after years of effort – didn’t require a breakthrough, or additional brainpower,

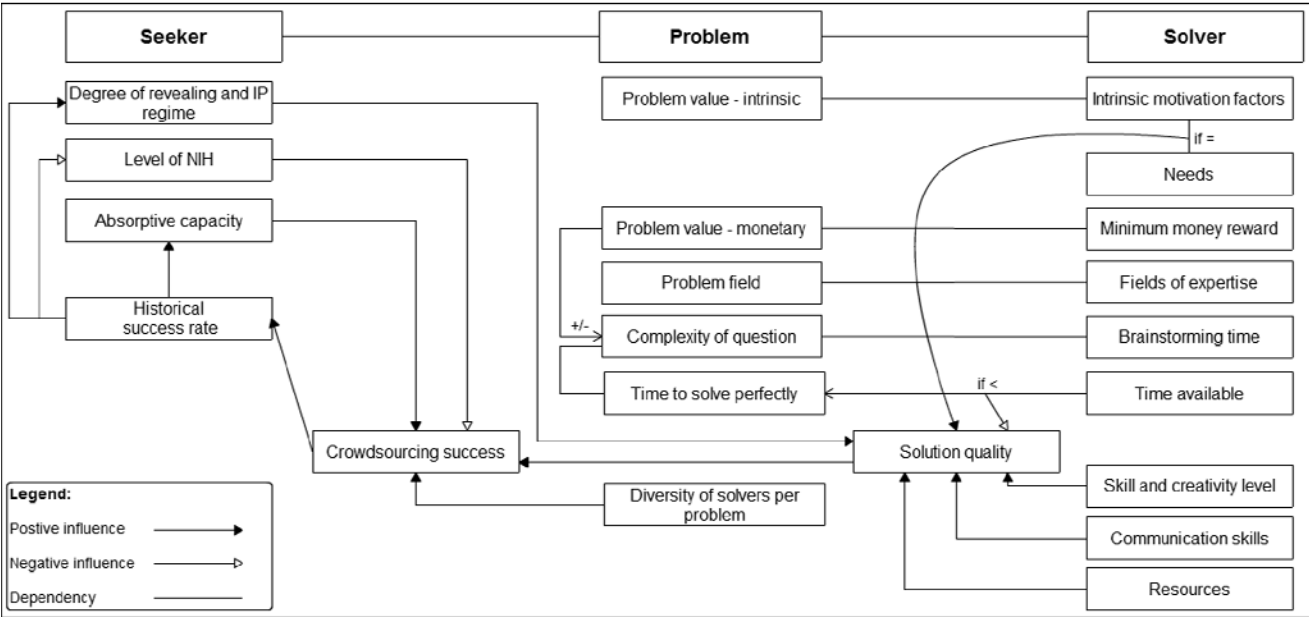


Figure 1 - Overview of simulation parameters described in this paper and their interactions

or a more talented scientist's attention; they just needed a diverse enough set of minds to have a go at them.

**Complexity of question.** This parameter indicates how complex a question is and the time investment needed by a solver. It also influences how well a solver can understand the question and how much time the solver needs to brainstorm a potential solution. (Benkler 2006) defines the benefits of dividing a problem to decrease its complexity as the “property of a project that describes the extent to which it can be broken down into smaller [...] modules that can be independently produced before they are assembled into a whole. [...] While creative capacity and judgment are universally distributed in a population, available time and attention are not.” (Schenk and Guittard 2009) differentiates between routine tasks Crowdsourcing and complex tasks Crowdsourcing: “Routine tasks Crowdsourcing seeks a number of complementary contributions necessary for the construction of data and information bases. Complex tasks Crowdsourcing follows a diametrically opposed pattern [...]”

**Time to solve perfectly.** This parameter is used in the simulation to determine the time a solver would need to solve a problem perfectly. However, it is not known to seekers or solvers, but only to the simulator.

**Diversity of solvers per problem.** This parameter combines the diversity of solvers (backgrounds, age, etc.) that worked on the solved problem. A higher diversity grade will generate a better winning solution (see “State of the art”).

**Solver Parameters**

The solver parameters include parameters for the members of the crowd and the submitted solutions (if any).

In a Crowdsourcing scenario, pedigree, race, gender, age, level of expertise and similar are not relevant (on most platforms, the participants are anonymous to others). Such typical moderating variables in team and sociological studies are not relevant to Crowdsourcing success.

**Intrinsic motivation factors.**

Successful Crowdsourcing involves satisfying very basic needs. (Bartl 2010) writes:

Drawing on a rich body of motivation research relevant motives are curiosity, self efficacy, skill development, information seeking, intrinsic playful task, recognition, altruism and community support, make friends, personal need/dissatisfaction or compensation and monetary rewards.

**Solution quality.** In the simulator, solution quality is influenced by the solver’s skill level, communication skills, resources (e.g., laboratory supplies), time available and the seeker’s degree of revealing (accuracy and background information in problem description). Examples of quality are, e.g., enhanced technical performance, lower cost, good reliability, contribution to the research question or uniqueness. Seekers judging the solutions are regarded as “satisficers” (Simon 1996).

**Fields of expertise.** Every agent has a set of skills in one or more fields of expertise, influencing both problem selection and solution success potential. However, the solver does not necessarily need to have a field of expertise that matches the problem field exactly (see explanation of problem field, above). A solver with high general skill and creativity levels is able to pick problems outside his or her fields of expertise. The “skills” parameter includes the solvers’ private information, as defined by von Hayek.

**Brainstorming time.** Indicates the total time a potential solver takes to understand and respond to a posted problem.

**Time available.** Analogous to computer programs like SETI@home that use collective CPU spare cycles for supercomputing and hence leverage the power of the network, crowd members generally contribute to posted problems during their “spare cycles”, their downtime and energy not claimed by work or family obligations (Howe 2008), hence every simulator agent has a defined time available.

**Skill and creativity level.** This parameter shows a solver’s ability to solve a problem well. With a skill and creativity level above a certain threshold, the solver is also able to select and work on problems outside the field of expertise.

**Communication skills.** A solver with a higher (written) communication skill will have a greater chance of winning since she or he is better able to describe the idea/solution. In addition, this parameter positively affects the communication with other crowd members (if enabled).

**Resources.** Comprises all relevant resources (infrastructure, tools etc.) except for time and money that an agent has at hand and that are relevant for the chosen problem.

**Needs.** An agent with a need related to a field or type of problem more likely picks that problem and, if working on the problem successfully (resources, skills, time etc.), has a higher likelihood of delivering a superior solution. (Putnam 2000) and several others found that social innovation often occurs in response to social needs and that market pull (identifying and understanding users’ needs) is substantially more important to product success than technology push.

#### Global Parameters and Further Comments

For this simulator version, we assume a “closed” Crowdsourcing system, i.e., members of the crowd cannot see solutions already submitted by other members.

**Acquaintances.** The set of agents (seekers and solvers) starts with a set of agents the agent in question knows from “earlier times”. With every Crowdsourcing interaction, the set grows. “Old acquaintances” might be forgotten over time.

**Goal.** Every agent has a “goal”. For seekers, this is usually “maximization of Crowdsourcing success”. Solvers have all kinds of goals, including maximizing an intrinsic motivation factor or monetary income.

**Direct communication.** A global parameter that toggles whether seekers and solvers can directly communicate after the solver has picked a problem. The parameter simulates anonymity that is ensured on many Crowdsourcing platforms. A solution delivered by a solver communicating with the seeker increases the chance of winning (better understanding of the problem and its circumstances and hence higher quality solution). Communication with other crowd members decreases the solution quality due to reduced independence and diversity.

Whenever a relation is needed between parameters and there is no empirical evidence available, we use a “power law” according to (Mandelbrot and Hudson 2008) including the special case of the “1:10:89” rule: for every 100 people on a given site, 1 will create something, 10 will vote on what he or she created; the remaining 89 will consume the creation. “Super contributors”, between 1% and 2.5% of all solvers, depending on the platform, are usually responsible for a large share of crowdsourced data and knowledge collections.

## Simulator Set-up

(Wooldridge 2008) writes: “the steady move away from machine-oriented views of programming toward concepts and metaphors that more closely reflect the way in which we ourselves understand the world” is an ongoing trend. Further, he says agent-based solutions are appropriate when “the environment is open, or at least highly dynamic, uncertain or complex” and “agents are a natural metaphor”. Our highly scalable simulator conforms to these prerequisites.

### The Jade Framework Used for the Simulator

Jade (“Java Agent DEvelopment Framework”) was developed in 2000 as “an enabling technology, a middleware for the development and run-time execution of peer-to-peer applications which are based on the agents paradigm” (Bellifemine et al. 2003).

Agents, as defined for this kind of multi-agent programming, are autonomous, proactive and social peers that are provided interoperation capabilities by the framework. Jade provides the programmer with the capabilities to create agents that are loosely coupled and come with a fully enabled asynchronous messaging system between all actors.

Further points for selecting Jade over other frameworks were its interoperability by being compliant with the FIPA (“Foundation for Intelligent Physical Agents,”) specifications, its open source license, the large programmer community, the amount of documentation available and the great scalability.

### Design

The simulator models both seekers and solvers involved in a typical Crowdsourcing context. The current version does not include the potential intermediary as an additional type of agent: The Jade Framework provides a “Yellow Pages” agent that is used as a “broker” and matches solvers with problems.

Solvers are able to search the yellow pages as a directory for problems matching their given skills. The seeker and solver can then start communicating with each other directly, if allowed.

After toggling global conditions, users define value ranges for the active parameters and choose which variable is the dependent variable for this simulation cycle. The user may also choose to load own data (the ranges are then discretized by the software) or select from a given set of probability distributions. Figure 2 shows a screenshot. The output variables can be aggregated over several cycles and used for sensitivity analyses and other (e.g., statistical) evaluations. For technical and implementation details please contact the second author.

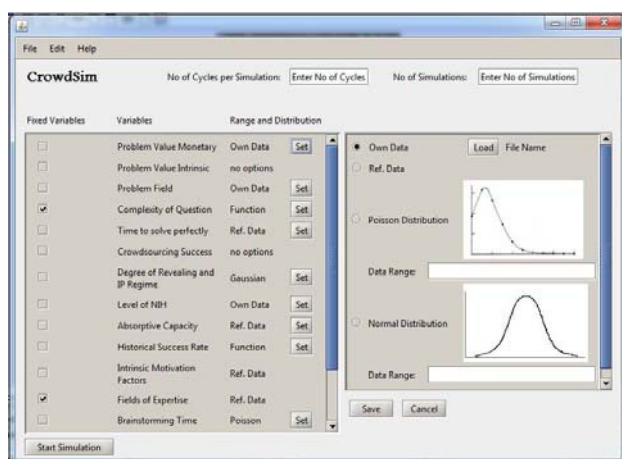


Figure 2 - Screenshot of parameter selection and data initialization screen

## Discussion and use cases

The simulator's robustness is currently being increased by incorporating data from Crowdsourcing in two domains: Corporate R&D and basic science as conducted at research universities. To this end, we will use data collected in the studies presented in (Lakhani et al. 2006) and (Buecheler et al. 2010). In addition, we have just started a collaboration with the largest Swiss Open Innovation site, "atizo.com" to gather more data from daily interactions. As the following three use cases show, the simulation is benefiting both researchers and practitioners:

### Use Case 1: Scientific Inquiry: Analogies to Biology

Scientists may (and will) use the simulator for confirming observations from biology in a complex network of irrational human agents. E.g., we hypothesize that solution development in open Crowdsourcing systems (as observed, e.g., in Mathwork's Matlab programming contests: Gulley 2004) resemble the developments of evolution in the sense that phases of regular, gradual evolution alternate with more punctuated sequences and stasis. (For a critical discussion see e.g., Smith 1988). In addition to these varying speeds, we expect to observe other phenomena like mass extinctions, co-evolution, and growing complexity of the behavior patterns and strategies that survive. (Emmeche 1994), (Lindgren 1992).

### Use Case 2: Scientific Inquiry: Crowdsourcing Dynamics

The simulator helps achieve three goals that (Axelrod 1997) has compiled: 1. Explore/describe: find basic system dynamics and qualitative interdependencies as well as quantitative correlations between (Crowdsourcing) parameters. For example, we hypothesize that the functions for resources, needs and fields of expertise have to be correlated to best approximate empirical outcomes and additionally communication skills need to be correlated with acquaintances. 2. Confirm/explain: use the simulator to explain agents' complex behavior and verify or falsify

assumptions while constantly ameliorating the simulator through empirical data. 3. Forecast/predict: consider complicated input variables and influencing factors to generate predictions, heuristics or narrower solution spaces.

### Use Case 3: Practical Use

Practitioners from the private sector (as well as scientists wishing to use Crowdsourcing as a tool, see Buecheler et al. 2010) may use the simulator for testing Crowdsourcing scenarios, parameter sensitivities and the optimal setting for their Crowdsourcing plans. Crowdsourcing consultancies, currently being founded all over the world, can use the simulator to test real-life settings. This not only helps increase effectiveness, but also communicates the value of Crowdsourcing and supports increasing the openness of the seeking organization.

## Conclusions

In contrast to simple existing models, the simulator allows the user to predict what is needed to achieve optimal Crowdsourcing results with the given resources, incorporating a large set of potential influences. In addition, the modular and extensible way the simulator is built enables the user to increase the accuracy and predictive power when scientists gain new empirical insights.

### Contributions by Authors

Most of the theoretical contributions were written by the first, third and fourth authors. The simulator design and details are mainly contributed by the second author.

## References

- Alonso, O., D. E. Rose, B. Stewart. 2008. Crowdsourcing for relevance evaluation. *Newsletter ACM SIGIR Forum* 42(2) 9–15.
- Axelrod, R. M. 1997. Advancing the art of simulation in the social science. *Complexity* 3 193–199.
- Bandte, H. 2007. *Komplexität in Organisationen. Organisationstheoretische Betrachtungen und agentenbasierte Simulation*. Deutscher Universitäts-Verlag / GWV Fachverlage GmbH Wiesbaden, Wiesbaden.
- Bartl, M. 2010. The Making-of Innovation » The Morphology of Co-Creation, <http://www.michaelbartl.com/co-creation/article/the-morphology-of-co-creation/> (July 12, 2010).
- Bellifemine, F., G. Caire, A. Poggi, G. Rimassa. 2003. Jade - A white paper. *exp* 3(3) 6–19.
- Benkler, Y. 2006. *The wealth of networks. How social production transforms markets and freedom*. Yale Univ. Press, New Haven.
- Brown, S. L., K. M. Eisenhardt. 1995. Product Development: Past Research, Present Findings, and Future Directions. *The Academy of Management Review* 20(2) 343–378.
- Buecheler, T., J. H. Sieg, R. M. Füchslin, R. Pfeifer. 2010. Crowdsourcing, Open Innovation and Collective Intelligence in the Scientific Method: A Research Agenda and Operational Framework. H. Fellermann, M. Dörr, M. M. Hanczyc, L. Ladegaard Laursen, S. Maurer, D. Merkle, P.-A. Monnard, K.



- Stoy, S. Rasmussen, eds. *Artificial Life XII. Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*. MIT Press, Cambridge, Mass., 679–686.
- Camazine, S., J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, E. Bonabeau. 2003. *Self-organization in biological systems*. Princeton University Press, Princeton, N.J.
- Chesbrough, H. W. 2003. *Open innovation. The new imperative for creating and profiting from technology*. Harvard Business School Press, Boston, Mass.
- Cohen, W. M., D. A. Levinthal. 1990. Absorptive Capacity: A New Perspective on Learning and Innovation. *Administrative Science Quarterly* 35(1, Special Issue: Technology) 128–152.
- Dahlander, L., D. Gann. 2010. How open is Innovation? *Research Policy*.
- David, P. A. 2003. The Economic Logic of “Open Science” and the Balance between Private Property Rights and the Public Domain in Scientific Data and Information: A Primer. *Stanford Institute For Economic Policy Research SIEPR Disc. Paper No. 02-30*.
- Dorigo, M., T. Stützle. 2004. *Ant colony optimization*. MIT Press, Cambridge, Mass.; London.
- Emmeche, C. 1994. *The garden in the machine. The emerging science of artificial life*. Princeton Univ. Press, Princeton, N.J.
- Giles, J. 2005. Internet encyclopedias go head to head. *Nature* 438 900–901.
- Granovetter, M. S. 1973. The Strength of Weak Ties. *The American Journal of Sociology* 78(6) 1360–1380.
- Gulley, N. 2004. In praise of tweaking: a wiki-like programming contest. *ACM Interactions* 11 18–23.
- Heller, M. A., R. S. Eisenberg. 1998. Can Patents Deter Innovation: the Anticommons in Biomedical Research. *Science* 280 698–701.
- Henkel, J. 2006. Selective Revealing in Open Innovation Processes: The Case of Embedded Linux. *Research Policy* 35(7) 953–969.
- Howe, J. 2008. *Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business*. Crown Publishing, New York.
- Katz, R., T. J. Allen. 2007. Investigating the Not Invented Here (NIH) syndrome: A look at the performance, tenure, and communication patterns of 50 R&D Project Groups. *R&D Mgmt* 12(1) 7–20.
- Kazman, R., H.-M. Chen. 2009. The Metropolis Model: A New Logic for Development of Crowdsourced Systems, <http://cacm.acm.org/magazines/2009/7/32094-the-metropolis-model-a-new-logic-for-development-of-crowdsourced-systems/fulltext> (April 5, 2010).
- Krause, J., G. D. Ruxton, S. Krause. 2010. Swarm intelligence in animals and humans. *Trends in Ecology & Evolution* 25(1) 28–34.
- Kittur, Aniket; Chi, Ed H.; Bongwon, Suh (2008): Crowdsourcing user studies with Mechanical Turk. In Mary Czerwinski, Arnie Lund: *Proceedings of the twenty-sixth annual SIGCHI Conference*. April 5-10, 2008, Florence, Italy. New York, N.Y.: ACM Press.
- Lakhani, K. R., L. B. Jeppesen, P. A. Lohse, J. A. Panetta. 2006. The Value of Openness in Scientific Problem Solving. *Harvard Business School*(HBS Working Paper Number: 07-050).
- Lee, S., B. Bozeman. 2005. The Impact of Research Collaboration on Scientific Productivity. *Social Studies of Science* 35(5) 673–702.
- Leimeister, J., M. Huber, U. Bretschneider, H. Krcmar. 2009. Leveraging Crowdsourcing: Activation-Supporting Components for IT-Based Ideas Competition. *Journal of Management Information Systems* 26(1) 197–224.
- Lindgren, K. 1992. Evolutionary phenomena in simple dynamics. C. G. Langton, C. Taylor, J. D. Farmer, S. Rasmussen, eds. *Artificial life II. Proceedings of the Workshop on Artificial Life, held February, 1990 in Santa Fé, New Mexico*. Addison-Wesley, Redwood City, Calif., 295–312.
- Mandelbrot, B. B., R. L. Hudson. 2008. *The (mis)behavior of markets. A fractal view of risk, ruin, and reward*, 1st ed. Profile Books, London.
- Mitleton-Kelly, E. 2003. Ten principles of complexity and enabling infrastructure. E. Mitleton-Kelly, ed. *Complex systems and evolutionary perspectives on organisations. The application of complexity theory to organisations*. Pergamon, Oxford, 23–50.
- Muhdi, L., M. Daiber, S. Friesike, R. Boutellier. 2010. Crowdsourcing: an alternative idea generation approach in the early innovation process. *The proceedings of the XXI ISIPIM Conference. The dynamics of Innovation*, [14 pages].
- Page, S. E. 2008. *The difference. How the power of diversity creates better groups, firms, schools, and societies*, 3rd ed. Princeton Univ. Press, Princeton, NJ.
- Putnam, R. D. 2000. *Bowling alone. The collapse and revival of American community*. Simon & Schuster, New York, NY.
- Robu, V., H. Halpin, H. Shepherd. 2009. Emergence of consensus and shared vocabularies in collaborative tagging systems. *ACM Transactions on the Web (TWEB)* 3(4) Article 14.
- Schelling, T. C. 1960. *The strategy of conflict*. Harvard Univ. Press, Cambridge, Mass.
- Schenk, E., C. Guittard. 2009. Crowdsourcing: What can be Outsourced to the Crowd, and Why? HAL: Sciences de l'Homme et de la Société. [archives-ouvertes.fr](http://archives-ouvertes.fr).
- Sieg, J. H., M. W. Wallin, G. von Krogh. 2011. Managerial challenges in open innovation: A study of innovation intermediation in the chemical industry. *R&D Management*(IN PRESS).
- Simon, H. A. 1996. *The sciences of the artificial*, 3rd ed. MIT Press, Cambridge, Mass.
- Smith, J. M. 1988. Punctuation in Perspective. *Nature*(332) 311–312.
- Surowiecki, J. 2004. *The wisdom of crowds. Why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations*, 1st ed. Doubleday, New York.
- Teece, D. J. 1986. Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. *Research Policy* 15(6) 285–305.
- von Ahn, Luis (2006): Games with a Purpose. *Computer* 39(6) 92–94.
- von Hayek, F. A. 1945. The Use of Knowledge in Society. *American Economic Review* 519–530.
- von Hippel, E. 1994. "Sticky Information" and the Locus of Problem Solving: Implications for Innovation. *Management Science* 40(4) 429–439.
- von Hippel, E., G. von Krogh. 2003. Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science. *Organization Science* 14(2) 209–223.
- Wooldridge, M. J. 2008. *An introduction to multiagent systems*. Wiley, Chichester.

# Critical Mutation Rate Has an Exponential Dependence on Population Size

Alastair Channon<sup>1</sup>, Elizabeth Aston<sup>1</sup>, Charles Day<sup>1</sup>, Roman V. Belavkin<sup>2</sup> and Christopher G. Knight<sup>3</sup>

<sup>1</sup> Research Institute for the Environment, Physical Sciences and Applied Mathematics, Keele University, ST5 5BG, UK

<sup>2</sup> School of Engineering and Information Sciences, Middlesex University, London, NW4 4BT, UK

<sup>3</sup> Faculty of Life Sciences, The University of Manchester, M13 9PT, UK

{a.d.channon,e.j.aston,c.r.day}@epsam.keele.ac.uk, r.belavkin@mdx.ac.uk, chris.knight@manchester.ac.uk

## Abstract

Populations of individuals exist in a wide range of sizes, from billions of microorganisms to fewer than ten individuals in some critically endangered species. In any evolutionary system, there is significant evolutionary pressure to evolve sequences that are both fit and robust; at high mutation rates, individuals with greater mutational robustness can outcompete those with higher fitness, a concept that has been referred to as survival-of-the-flattest. Previous studies have not found a relationship between population size and the mutation rate that can be tolerated before fitter individuals are outcompeted by those that have a greater mutational robustness. However, using a genetic algorithm with a simple two-peak fitness landscape, we show that the mutation rates at which the high, narrow peak and the lower, broader peak are lost for increasing population sizes can be approximated by exponential functions. In addition, there is evidence for a continuum of mutation rates representing a transition from survival-of-the-fittest to survival-of-the-flattest and subsequently to the error catastrophe. The effect of population size on the critical mutation rate is shown to be particularly noticeable in small populations. This provides new insight into the factors that can affect survival-of-the-flattest in small populations, and has implications for populations under threat of local extinction.

## Introduction

Biological population sizes can range from small numbers of individuals to very large numbers of individuals. For example, RNA viruses can reach population sizes of around  $10^{10}$  in a short amount of time (Comas et al., 2005), whereas some animal species may exist in populations consisting of only hundreds or even fewer than ten individuals in some critically endangered species saved on the brink of extinction. A population of genomes constantly evolves through the processes of mutation, recombination (in sexual reproduction), selection and genetic drift (Hartl and Clark, 2007). Population dynamics can be modelled *in silico* using genetic algorithms, in which populations of sequences are allowed to undergo mutation, recombination and selection at specified rates; studies can be done in a controlled environment within time-frames not possible in many natural biological systems, producing results that are comparable both to theory and to experimental results in microorganisms.

In any evolutionary system, including genetic algorithms and natural biological systems, there is significant evolutionary pressure to evolve sequences that are both fit and robust (Jones and Soule, 2006). Robustness is defined as the average effect of a specified perturbation (such as a new mutation) on the fitness of a specified genotype (Masel and Trotter, 2010). The more robust a genotype, the smaller the effect of mutation on fitness; in systems with high levels of mutation, robustness can reduce the negative effects of deleterious mutation. Smaller populations are more susceptible to loss of fitness through random genetic drift (Comas et al., 2005; Hartl and Clark, 2007). Therefore it is expected that population size should influence the size of mutation rate that can be tolerated before fitter individuals are outcompeted by those with a greater mutational robustness.

## Mutational Robustness and Survival-of-the-flattest

The concept of a fitness landscape was introduced by Wright (1932) and later combined with the notion of sequence space by Eigen and Schuster (1979). Each sequence in sequence space has a fitness value, which represents its relative replication capacity (Domingo and Wain-Hobson, 2009). Fitness landscapes are sometimes considered to resemble mountain ranges, with the fittest sequences at the peaks. However, the concept requires a more careful interpretation in high dimensional sequence spaces with low alphabet size, such as nucleic acids, which have an alphabet size of four (in that they are sequences consisting of four possible units, A, C, G and T). For example, the space of  $N$ -length binary sequences is an  $N$ -dimensional hypercube rather than a 3-dimensional Euclidean geometry, and can only be represented as such by use of a reductive transform between the two spaces. Exploration of sequence space is done through evolution by mutation, recombination and selection in accordance to the fitness landscape. Selection increases the frequencies of the fittest sequences, while mutation introduces variation, often at a cost to individual fitness. The balance between these two forces is referred to as the mutation-selection balance (Kimura and Maruyama, 1966; Bull et al., 2005). A popula-

tion in mutation-selection balance will tend to cluster around the fitness peaks and form what is known as a quasispecies (Eigen and Schuster, 1979; Bull et al., 2005; Nowak, 2006).

In a landscape with a single fitness peak, a quasispecies is able to maintain its position surrounding the top of the peak so long as the mutation rate does not exceed a particular rate known as the *error threshold*. Above this threshold, there is an *error catastrophe* and the population delocalizes across sequence space (Tannenbaum and Shakhnovich, 2004; Bull, 2005; Nowak, 2006; Takeuchi and Hogeweg, 2007; Domingo and Wain-Hobson, 2009; Schuster, 2009; Tejero et al., 2011).

The concept of error threshold was introduced in Eigen et al. (1988) and later in Nowak and Schuster (1989) based on the quasispecies equation:

$$\dot{x}_i = \sum_{j=1}^m x_j f_j q_{ji} - \phi x_i$$

Here,  $x_i$  is the frequency of genotype number  $i$ , where  $i \in [1, \dots, \alpha^n]$ ,  $\alpha$  is the alphabet size,  $n$  is the length of sequences,  $\sum x_i = 1$ ,  $f_j$  is fitness (selection),  $\phi = \sum x_i f_i$  is the average fitness, and  $q_{ji}$  is a transition probability (mutation). The derivative in time is denoted  $\dot{x}$ , and there are  $m$  genetic sequences.

Selection and mutation provide two forces (or pressures) on the population, and they can be combined into one matrix ( $w_{ji} = f_j q_{ji}$ ) (see Nowak (2006), p. 35). Selection draws the population closer to the highest fitness, while mutation is usually assumed to have deleterious effect due to which the population drifts away from the highest fitness. Generally, the population converges to a stable (equilibrium) state that is defined by an eigenvector of the mutation-selection matrix ( $w_{ji}$ ). This eigenvector corresponds to the largest eigenvalue of ( $w_{ji}$ ), which is the average fitness  $\phi$ .

The idea of an error threshold is based on the existence of a mutation-selection balance when the effect of mutation does not exceed that of the selection pressure. The corresponding value of the mutation rate is referred to as the error threshold, and it is the maximal mutation rate that allows a population to stay centred ‘around’ the fitness peak.

However, in landscapes with more than one peak, there may also be one (or more) *critical mutation rates* at which the population loses its ability to localize to fitter peaks, while potentially retaining its ability to remain on lower, flatter peaks (Wilke et al., 2001; Tannenbaum and Shakhnovich, 2004; Comas et al., 2005; Wilke, 2005). This represents a phase transition from survival-of-the-fittest to survival of those individuals with greater mutational robustness, a concept referred to as survival-of-the-flattest (Wilke et al., 2001; Bull et al., 2005; Comas et al., 2005; Wilke, 2005; Sanjuán et al., 2007; Sardanyés et al., 2008; Tejero et al., 2011). This concept is based on the idea that at low mutation rates, selection favours individuals in a quasispecies that reside at peaks

with higher fitness, even if the peaks are steep and narrow, due to the rarity of mutations that push individuals off the peaks (Lenski et al., 2006). In contrast, at high mutation rates, selection favours individuals that reside at peaks less likely to result in off-peak mutations: individuals located in flatter regions of the fitness landscape are less likely to suffer large reductions in fitness compared with those that may be initially fitter but reside in parts of the landscape with steeper peaks. Individuals that are part of a neutral network (Kimura, 1983), in that they are surrounded by other individuals with equivalent fitness, are said to be mutationally robust (Bull et al., 2005; Bornberg-Bauer and Kramer, 2010; Wilke, 2001a; Wilke, 2001b); their fitness will be less sensitive to mutation than individuals that are not well connected.

Survival-of-the-flattest has been observed in digital organisms (Wilke et al., 2001; Sardanyés et al., 2008), theoretically (Wilke, 2001a; Sardanyés et al., 2008), in simulated RNA evolution (Wilke, 2001b), and in RNA viruses (Sanjuán et al., 2007). In addition, evolution of mutational robustness has been observed in simulated RNA evolution (van Nimwegan et al., 1999) and in laboratory protein evolution experiments (Bloom et al., 2007). Both van Nimwegan et al. (1999) and Bloom et al. (2007) place an emphasis on the degree of polymorphism in the population, suggesting that highly polymorphic populations are more likely to spread across many nodes of a neutral network (each corresponding to a genotype), concentrating at highly connected parts; individuals at highly connected nodes have greater robustness to mutation, which they pass on to the next generation. Robustness will evolve in any population where the product of the population size and frequency of mutation per sequence per generation is sufficiently large ( $>1$ ). Krakauer and Plotkin (2002) refer to flat landscapes as redundant, and steeper landscapes as antiredundant. They suggest that both in theory and in individual-based stochastic simulations, redundancy increases the mean fitness in small populations as it masks mutations that arise due to mutational drift. However, large populations are less affected by drift, and so are more able to occupy high-fitness peaks in sharp landscapes.

Wilke (2001b) ran simulations with population sizes as low as 100 and noted “that for very small populations, the predictive value of the differential equation approach diminishes”. Later Wilke noted that his results agreed with Comas et al. (2005) in finding “that population size played only a minor role in determining the position of the critical mutation rate” (Wilke, 2005), within the context of their experiments. Comas et al. used population sizes as low as 250 and concluded “that the critical mutation rate was independent of population size” (Comas et al., 2005) despite the fact that there did appear to be some correlation for certain cases.

Jones and Soule (2006) determined that the role of genetic robustness in evolution differs significantly depending on whether it is a generational or steady state genetic algorithm that is being used. In a steady state algorithm, only

a few individuals are replaced at a time, as opposed to a generational algorithm which replaces the entire population at once. Many studies that have confirmed the notion of survival-of-the-fittest have used generational models, such as Wilke et al.'s (2001) evolution of digital organisms in Avida, and Krakauer and Plotkin's (2002) study of redundancy and antiredundancy (Jones and Soule, 2006). Jones and Soule suggest that for evolutionary dynamics experiments, the class of algorithm used can have a significant effect on the observed outcome. They point to steady state algorithms as being of particular interest to the artificial life community, as natural evolution resembles the action of a steady-state-like algorithm: evolution in biological systems does not usually follow the generational approach of evolving every individual in the population synchronously.

However, the problem with steady state algorithms is that they typically allow individuals to survive on fitness peaks indefinitely. This is not a realistic property when modelling evolutionary dynamics. A preferable approach is to use a generational genetic algorithm which retains the key features of steady state evolution: fitness rank-based selection and a degree of asynchronicity. It should be noted that fitness in this sense refers to a score assigned to each individual based on a given fitness function, as opposed to the biological definition of fitness as a measure of replication rate; the exact fitness values used are unimportant as it is relative fitness that determines which individuals are selected. Rank-based selection (the assignment of reproductive fitness rates according to fitness score rank) overcomes the scaling problems of fitness score proportional selection (the assignment of reproductive fitness rates in proportion to fitness score), so creating a general model from a specific fitness score landscape such as that in figure 1, while retaining the key property that sequences with higher fitness scores have (probabilistically) more offspring than those with lower scores. This approach also allows for the existence of a critical mutation rate: with a standard steady state algorithm, always retaining the fittest individual prevents the population from ever losing the highest current peak.

### Simulation Model

An individual sequence consists of a string of characters drawn from an alphabet of size 4 (which can be thought of as, for example, A/C/G/T or 0/1/2/3) with a fixed length of 30. In each step of the algorithm, three individual sequences are selected at random from the population. Two of the three selected individuals are chosen as parents in a crossover which replaces the third individual with the resulting child. The child is then subject to one round of point mutation (to a *different* base) at a given per-base mutation rate. The individual to be replaced is decided each time based on the fitnesses of the three selected individuals: there is an equally small chance of either of the two fittest of the three being replaced (25%), and a larger chance of replacing the

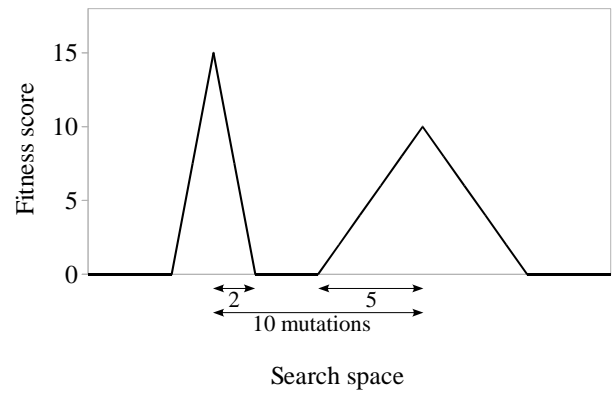


Figure 1: Two-peak fitness landscape, with one narrow peak of high fitness (peak 0), and one broader, flatter peak of lower fitness (peak 1).

least fit (50%). This process continues until each individual in the population has been chosen exactly once; this represents one generation, and ensures that there is no chance of any individual avoiding being chosen and so remaining static in the landscape. The fitness of each individual sequence is evaluated based on a two-peak fitness landscape with one narrow peak of high fitness (peak 0), and a broader, flatter peak with lower fitness (peak 1) (figure 1). Peak 0 has a maximum fitness score of 15 and a radius (Hamming distance from top-of-peak to zero fitness score) of 2; peak 1 has a maximum fitness score of 10 and a radius of 5, with its top chosen as an arbitrary point (fixed throughout evolution) with a Hamming distance of 10 from the top of peak 0. Individuals are allowed to move anywhere on the slopes, or in between the peaks. This is a simple landscape in which survival-of-the-fittest can occur. The effect of mutation on fitness is smaller within peak 1 than within peak 0; individuals located on peak 1 will have higher mutational robustness compared with those located on peak 0.

Following the experimental procedure designed by Wilke et al. (2001) (and used by Comas et al., 2005) we initialized half of the population of sequences to be on top of the high, narrow peak, and the other half to be on top of the lower, flatter peak. This procedure prevents initialization bias between peaks. The simulation was run for 10,000 generations, and the number of generations it took to first lose each peak was recorded (where a peak was considered to be lost when there were no individuals present anywhere in its range). If a peak was never lost within the 10,000 generations, a value of -1 was recorded. A range of per-base mutation rates was tested for a range of population sizes. The simulation was run 2,000 times for each combination of mutation rate and population size. The mutation rate by which 95% of the runs had lost each peak was recorded, where a peak was considered to have not ever been lost only if there were individuals remaining on it at the end of the 10,000 generations.

Population size (m)	Observed $\mu_0$	Stretched Exponential $\epsilon_0$	Difference $\delta_0$ ( $\mu_0 - \epsilon_0$ )	Difference/Stretched Exp. ( $\delta_0/\epsilon_0$ )
10	0.150%	0.150%	0.000%	0.1%
20	0.550%	0.554%	-0.004%	-0.8%
30	0.750%	0.742%	0.008%	1.0%
40	0.850%	0.853%	-0.003%	-0.4%
50	0.925%	0.926%	-0.001%	-0.1%
60	0.975%	0.978%	-0.003%	-0.3%
70	1.025%	1.017%	0.008%	0.8%
80	1.050%	1.046%	0.004%	0.4%
90	1.065%	1.070%	-0.005%	-0.4%
100	1.080%	1.089%	-0.009%	-0.8%
200	1.170%	1.172%	-0.002%	-0.2%
300	1.200%	1.197%	0.003%	0.3%
400	1.210%	1.207%	0.003%	0.2%
500	1.215%	1.212%	0.003%	0.2%
600	1.220%	1.215%	0.005%	0.4%
700	1.225%	1.217%	0.008%	0.7%
800	1.225%	1.218%	0.007%	0.6%
900	1.210%	1.219%	-0.009%	-0.7%
1000	1.205%	1.219%	-0.014%	-1.2%

Table 1: Mutation rate  $\mu_0$  by which 95% of runs lost peak 0.

## Results

The results (figure 2, and tables 1 and 2) show that population size affects the size of mutation rate required for the predominant outcome of the runs to shift from survival-of-the-fittest to survival-of-the-flattest, and that this is particularly noticeable in populations with 100 individuals or less. Similarly, the size of mutation rate required for approximately 95% of the runs to have lost both peaks also has a dependence on population size. The results of the simulation can be approximated by a simple exponential function:  $y = A - B m^C$  for some values of the parameters  $A$ ,  $B$  and  $C$ , where  $m$  is population size. However, they are more closely fitted by a stretched exponential function:  $y = A - B * e^{-((m/C)^D)}$ .

As opposed to there being instantaneous transitions from survival-of-the-fittest to survival-of-the-flattest and to the error catastrophe, at discrete mutation rates, there appear to be gradual transitions in which there are shifts in tendency from the first to the second, and from the second to the third. The mutation rate corresponding to 95% of the runs having lost the high, narrow peak (peak 0) within 10,000 generations marks a point at which the former transition (from survival-of-the-fittest to survival-of-the-flattest) is essentially complete. This can be considered as a critical mutation rate. For a population of 100 individuals, this is at a per-base mutation rate of approximately 1.08% (table 1). Figure 3(a) shows the number of generations taken to lose each peak at this mutation rate, for each of the 2,000 runs with population size 100. Just 52% of these runs lost peak 1 within the dura-

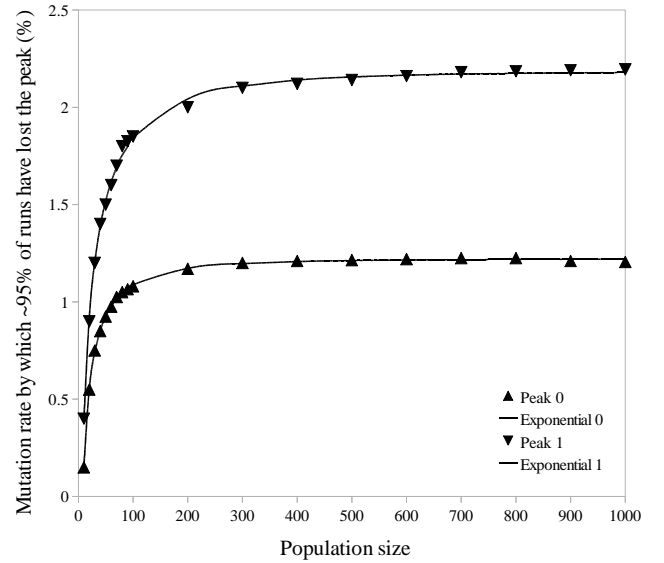


Figure 2: The results of the simulation for both peak 0 (high, narrow peak) and peak 1 (lower, flatter peak) can be approximated by an exponential function, where  $y = A - B * e^{-((m/C)^D)}$  (with  $m$  being population size). The parameters obtained by curve-fitting using a least squares method were, for peak 0:  $A = 1.221\%$ ,  $B = 7.001\%$ ,  $C = 1.440$ ,  $D = 0.3250$ , and for peak 1:  $A = 2.184\%$ ,  $B = 5.438\%$ ,  $C = 7.721$ ,  $D = 0.3978$ .

Population size (m)	Observed $\mu_1$	Stretched Exponential $\epsilon_1$	Difference $\delta_1$ ( $\mu_1 - \epsilon_1$ )	Difference/Stretched Exp. ( $\delta_1/\epsilon_1$ )
10	0.400%	0.389%	0.011%	2.8%
20	0.900%	0.921%	-0.021%	-2.3%
30	1.200%	1.206%	-0.006%	-0.5%
40	1.400%	1.390%	0.010%	0.7%
50	1.500%	1.520%	-0.020%	-1.3%
60	1.600%	1.617%	-0.017%	-1.0%
70	1.700%	1.692%	0.008%	0.4%
80	1.800%	1.753%	0.047%	2.7%
90	1.825%	1.802%	0.023%	1.3%
100	1.850%	1.843%	0.007%	0.4%
200	2.000%	2.043%	-0.043%	-2.1%
300	2.100%	2.109%	-0.009%	-0.4%
400	2.120%	2.140%	-0.020%	-0.9%
500	2.140%	2.156%	-0.016%	-0.7%
600	2.160%	2.165%	-0.005%	-0.2%
700	2.180%	2.171%	0.009%	0.4%
800	2.185%	2.174%	0.011%	0.5%
900	2.190%	2.177%	0.013%	0.6%
1000	2.195%	2.179%	0.016%	0.8%

Table 2: Mutation rate  $\mu_1$  by which 95% of runs lost peak 1.

tion of the simulation (compared to 95% for peak 0). At this mutation rate, early loss of peak 0 appears to be a condition for survival-of-the-flattest. Loss of peak 0 is then followed by one of two events: either peak 1 is lost relatively quickly (within 200 generations) or it is maintained for the duration of the simulation. The fate of the population after loss of peak 0 is therefore dependent on whether or not it is able to quickly converge on peak 1. Figure 3(a) shows (at this mutation rate) that when peak 0 is not lost early, the number of generations taken to lose peak 0 is distributed approximately evenly up to 10,000 generations.

The mutation rate corresponding to 95% of the runs having lost the lower, flatter peak (peak 1) within 10,000 generations marks a point at which the latter transition (from survival-of-the-flattest to the error catastrophe) is essentially complete. This can be considered as another critical mutation rate (or the error threshold). For a population of 100 individuals, this is at a per-base mutation rate of approximately 1.85% (table 2). Figure 3(b) shows the number of generations taken to lose each peak at this mutation rate, for each of the 2,000 runs with population size 100. It is an apparent reversal of figure 3(a) but with 100% of the runs having lost peak 0 within 200 generations. The population has almost entirely lost the ability to localize to either peak.

## Discussion

At high mutation rates, individuals with greater mutational robustness can outcompete those with higher fitness. Previous studies have not found a relationship between population size and the critical mutation rate, at which there is

a phase transition from survival-of-the-fittest to survival-of-the-flattest (Comas et al., 2005). However, the results of the current study suggest that population size does have an effect on the size of mutation rate that can be tolerated before the population loses the fittest and the flattest peaks, and that this is particularly noticeable in populations with 100 individuals or less. As shown in figure 2, the size of mutation rate at which each peak is lost for increasing population sizes can be approximated by an exponential function. One possible reason for this is that small populations are more susceptible to stochastic variation due to random genetic drift (Comas et al., 2005; Hartl and Clark, 2007); small populations with relatively large genomes cannot explore the entire neutral space of the landscape. Consequently, quasispecies formation is difficult, and the fitness peaks may be more easily lost. The dramatic reduction in critical mutation rate observed for small populations has implications for local extinction events in which there is a significant drop in population size. Further work will be necessary to apply this result to populations under threat of local extinction.

The dynamics of finite populations have very different properties compared to those of infinite populations, for example non-zero probability of extinction. The latter can be a good approximation of the former if the size of populations is large. However, where a small population size is fundamental to the issue of concern, as with the relationship established empirically in this paper, and in any work on extinction events (zero population size), such approximations break down. This situation is similar to statistical mechanics, where systems of large numbers of particles

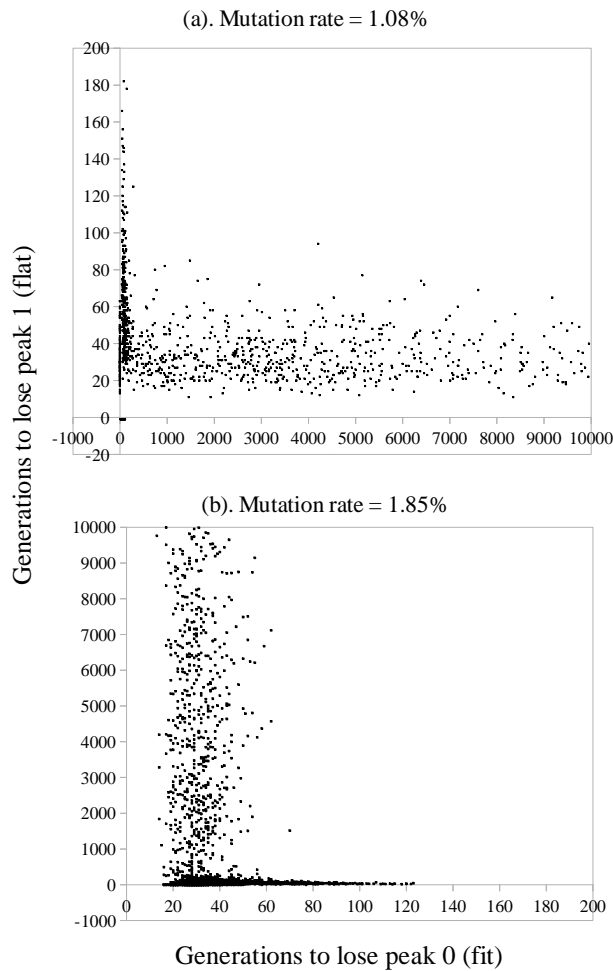


Figure 3: Transition from survival-of-the-fittest to survival-of-the-flattest and subsequently to the error catastrophe. Each point represents the number of generations it took to lose the high, narrow peak (peak 0) and the number to lose the lower, flatter peak (peak 1), in a single run of the GA for population size 100. Where a peak was not lost within 10,000 generations, a value of -1 was assigned for that particular run of the GA: all points on the negative side of either axis should be taken to have a higher value than 10,000. (a) The mutation rate by which 95% of the runs had lost peak 0 within the duration of the simulation; just 52% of these runs lost peak 1. This demonstrates that the transition from survival-of-the-fittest to survival-of-the-flattest is essentially complete. This can be considered as a critical mutation rate. (b) The mutation rate by which 95% of the runs had lost peak 1 within the duration of the simulation; 100% of these runs lost peak 0. This demonstrates that the transition from survival-of-the-flattest to the error catastrophe is essentially complete, with the population having almost entirely lost the ability to localize to either peak.

are approximated by laws derived for an infinite number of particles. The relation between the two is asymptotic and rooted in the law of large numbers. In fact, one can obtain equations for infinite populations from stochastic equations for finite populations by taking their expected value with respect to a probability measure on the population sizes  $m \in \{0, 1, \dots\}$ . The dynamics of finite populations can be described by stochastic differential equations. In particular, branching processes have been used to study the population dynamics of populations with variable (random) finite size (Jagers, 1975). The dynamics of finite populations have also been studied using the Moran process (Moran, 1962; Nowak, 2006). This work establishes an important empirical relationship between population size and critical mutation rate; the development of a corresponding theoretical model deserves further investigation.

Previous studies have defined the critical mutation rate to be the midpoint between the highest mutation rate at which there is survival-of-the-fittest, and the lowest mutation rate at which there is survival-of-the-flattest (Wilke et al., 2001; Comas et al., 2005). However, the results of this study clearly show that there is a transition from survival-of-the-fittest to survival-of-the-flattest and subsequently to the error catastrophe (figure 3).

## Conclusion

This study investigated whether or not there is a relationship between population size and the size of mutation rate that can be tolerated before fitter individuals are outcompeted by those that have a greater mutational robustness (the critical mutation rate). The results show that the sizes of mutation rate at which the high, narrow peak and the lower, flatter peak are lost for increasing population sizes can be approximated by an exponential function. The effect of population size on the size of mutation rate that can be tolerated before the population loses the fittest and the flattest peaks is particularly noticeable in small populations with 100 individuals or less. This provides new insight into the factors that can affect survival-of-the-flattest in small populations, and has implications for populations under threat of local extinction. Other factors, such as sequence length and distance between peaks, may well have a significant influence on both critical mutation rate and population sizes that can withstand specific rates of mutation. It will be beneficial to investigate this in the future, as well as to construct a theoretical model (whether based on differential equations or not) that can replicate the exponential relationship between critical mutation rate and population size, found here by experiment, for low population sizes.

In addition, there is clear evidence for a continuum of mutation rates representing a transition from survival-of-the-fittest to survival-of-the-flattest. This identifies a critical mutation rate by which the population has a 95% likelihood of losing the higher peak.

## Acknowledgments

This work was supported by EPSRC grant EP/H031936/1.

## References

- Bloom, J. D., Lu, Z., Chen, D., Raval, A., Venturelli, O. S., and Arnold, F. H. (2007). Evolution favours protein mutational robustness in sufficiently large populations. *BMC Biology*, 5:29.
- Bornberg-Bauer, E. and Kramer, L. (2010). Robustness versus evolvability: a paradigm revisited. *HFSP Journal*, 4(3-4):105–108.
- Bull, J. J., Meyers, L. A., and Lachmann, M. (2005). Quasispecies made simple. *PLoS Computational Biology*, 1(6):e61.
- Comas, I., Moya, A., and González-Candelas, F. (2005). Validating viral quasispecies with digital organisms: a re-examination of the critical mutation rate. *BMC Evolutionary Biology*, 5:5.
- Domingo, E. and Wain-Hobson, S. (2009). The 30th anniversary of quasispecies. *EMBO Reports*, 10:444–448.
- Eigen, M., McCaskill, J., and Schuster, P. (1988). Molecular quasispecies. *Journal of Physical Chemistry*, 92:6881–6891.
- Eigen, M. and Schuster, P. (1979). *The hypercycle*. Springer, New York.
- Hartl, D. L. and Clark, A. G. (2007). *Principles of population genetics*. Sinauer Associates, Inc., Sunderland, MA, 4th edition.
- Jagers, P. (1975). *Branching processes with biological applications*. Wiley.
- Jones, J. and Soule, T. (2006). Comparing genetic robustness in generational vs. steady state evolutionary algorithms. In *Proceedings of the 8th annual conference on genetic and evolutionary computation*, pages 143–149.
- Kimura, M. (1983). *The neutral theory of molecular evolution*. Cambridge University Press.
- Kimura, M. and Maruyama, T. (1966). The mutational load with epistatic gene interactions in fitness. *Genetics*, 54:1337–1351.
- Krakauer, D. C. and Plotkin, J. B. (2002). Redundancy, antiredundancy, and the robustness of genomes. *Proceedings of the National Academy of Sciences of the United States of America*, 99(3):1405–1409.
- Lenski, R., Barrick, J. E., and Ofria, C. (2006). Balancing robustness and evolvability. *PLoS Biology*, 4:2190–2192.
- Masel, J. and Trotter, M. V. (2010). Robustness and evolvability. *Trends in Genetics*, 26:406–414.
- Moran, P. A. P. (1962). *The Statistical Processes of Evolutionary Theory*. Clarendon Press, Oxford.
- Nowak, M. A. (1992). What is a quasispecies? *Trends in Ecology and Evolution*, 7:118–121.
- Nowak, M. A. (2006). *Evolutionary dynamics: Exploring the equations of life*. Harvard University Press.
- Nowak, M. A. and Schuster, P. (1989). Error thresholds of replication in finite populations: Mutation frequencies and the onset of Muller’s ratchet. *Journal of Theoretical Biology*, 137:375–395.
- Sanjuán, R., Cuevas, J. M., Furió, V., Holmes, E. C., and Moya, A. (2007). Selection for robustness in mutagenized RNA viruses. *PLoS Genetics*, 3(6):e93.
- Sardanyés, J., Elena, S. F., and Solé, R. V. (2008). Simple quasispecies models for the survival-of-the-flattest effect: The role of space. *Journal of Theoretical Biology*, 250:560–568.
- Schuster, P. (2009). Genotypes and phenotypes in the evolution of molecules. *European Review*, 17(2):281–319.
- Takeuchi, N. and Hogeweg, P. (2007). Error-threshold exists in fitness landscapes with lethal mutants. *BMC Evolutionary Biology*, 7:15.
- Tannenbaum, E. and Shakhnovich, E. I. (2004). Solution of the quasispecies model for an arbitrary gene network. *Physical Review E*, 70:021903.
- Tejero, H., Marin, A., and Montero, F. (2011). The relationship between error catastrophe, survival of the flattest, and natural selection. *BMC Evolutionary Biology*, 11:2.
- van Nimwegen, E., Crutchfield, J. P., and Huynen, M. (1999). Neutral evolution of mutational robustness. *Proceedings of the National Academy of Sciences of the United States of America*, 96:9716–9720.
- Wilke, C. O. (2001a). Adaptive evolution on neutral networks. *Bulletin of Mathematical Biology*, 63:715–730.
- Wilke, C. O. (2001b). Selection for fitness vs. selection for robustness in RNA secondary structure folding. *Evolution*, 55:2412–2420.
- Wilke, C. O. (2005). Quasispecies theory in the context of population genetics. *BMC Evolutionary Biology*, 5:44.
- Wilke, C. O. and Adami, C. (2003). Evolution of mutational robustness. *Mutation Research*, 522:3–11.



- Wilke, C. O., Wang, J. L., Ofria, C., Lenski, R. E., and Adami, C. (2001). Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, 412:331–333.
- Wright, S. (1932). The roles of mutation, inbreeding, cross-breeding, and selection in evolution. In *Proceedings of the Sixth International Congress on Genetics*, pages 355–366.

# Emergence of Temporal and Spatial Synchronous Behaviors in a Foraging Swarm

Sylvain Chevallier<sup>1</sup>, Nicolas Bredeche<sup>1</sup>, H       Paugam-Mois  <sup>1,2</sup> and Mich     Sebag<sup>1</sup>

<sup>1</sup>TAO/LRI, Univ. Paris-Sud, CNRS, INRIA, F-91405 Orsay, France

<sup>2</sup>LIRIS, Univ. Lyon 2, CNRS, F-69676 Bron, France

FirstName.Name@lri.fr

## Abstract

Biological populations often exhibit complex and efficient behaviors, where temporal and spatial couplings at the macro-scale population level emerge from interactions at the micro-scale individual level, without any centralized control. This paper specifically investigates the emergence of behavioral synchronization and the division of labor in a foraging swarm of robotic agents. A deterministic model is proposed and used by each agent to decide whether it goes foraging, based on local cues about its fellow ants' behavior. This individual model, based on the competition of two spiking neurons, results in a self-organized division of labor at the population level. Depending on the strength and occurrences of interactions among individuals, the population behavior displays either an asynchronous, or a synchronous aperiodic, or a synchronous periodic division of labor. Further, the benefits of synchronized individual behaviors in terms of overall foraging efficiency are highlighted in a 2D spatial simulation.

## Introduction

Nature displays fascinating examples of biological populations that achieve complex tasks without requiring any centralized control. How to efficiently achieve a distributed and decentralized control, a key issue for biological and artificial systems alike, is still far from being entirely elucidated (Camazine et al., 2001), although the interplay between the individual, micro-scale level and the population, macro-scale level has been extensively studied in the literature (see Beshers and Fewell (2001) for a survey).

This paper focuses on behavioral synchronization and the division of labor in a robotic swarm. On the biological and ethological side, behavioral synchronization and division of labor have been shown to enhance the adaptive value in various insect species such as spiders (Krafft and Pasquet, 1991), collembola (Leinaas, 1983), fireflies (Branham and Greenfield, 1996) and have also been observed in ants (Goss and Deneubourg, 1988; Cole, 1991). Experimental studies devoted to the foraging behavior in ant colonies suggest that synchrony might contribute to a better communication among agents (Bonabeau et al., 1998b), and significantly improves the foraging performance compared to asynchronous behaviors (Bonabeau et al., 1998a; Delgado

and Sol  , 2000). The emergence of synchrony is explained from both individual factors, e.g. internal individual mechanisms, and local interactions among individuals.

On the artificial and robotic side, the self-organized division of labor in an ant colony is nothing like easily mastered by a robotic swarm. Notably, in many cases the micro-scale models proposed in the ethology literature might exceed the plausible physical or cognitive resources of most simple agents (e.g. due to the required resources or the presence of random generators supporting stochastic models). The swarm robotics framework involves specific limitations; while it considers a large population<sup>1</sup>, power consumption remains a critical issue, entailing limited communication and computational abilities. Quite a few authors have been considering foraging robotic swarms in the last two decades (see Bayindir and Sahin (2007) for a survey), proposing hand-crafted architectures (be they bio-inspired (Labella et al., 2006; Panait and Luke, 2004) or not (Liu et al., 2007; Hauert et al., 2008)), or using evolutionary computation to optimize the individual decision model (Dorigo et al., 2005).

In the meanwhile, how to enforce the synchrony of individual behaviors has seldom be considered. Wischmann et al. (2006) and Hartbauer and Romer (2007) have investigated the use of coupled oscillator-based models, respectively considering an energy-foraging and a cleaning task. Taking inspiration from insect synchronous behavior such as chorusing male insects, both approaches illustrate how group synchronization can emerge from local communications. Trianni and Nolfi (2009) present a thorough study of swarm synchronization from the perspective of dynamic systems, notably using Evolutionary Computation to optimize efficient synchronization strategies.

Resuming an earlier work Chevallier et al. (2010), this paper presents a frugal model aimed at robotic swarm foraging, called *SpikeAnts*. This model, based on the coupling of two spiking neurons with different internal dynamics (Gerstner and Kistler, 2002), enables an individual

<sup>1</sup>Contrasting with early work on multi-robot systems; see (Parker, 2008) and references therein.

agent to decide whether it must go foraging based on local cues from other agents. Spiking neurons, now a well-established formalism, are known to display rich temporal dynamics and synchronization patterns (Paugam-Moisy and Bohte, 2011). From the individual decisions and interactions within *SpikeAnts*, synchronous and asynchronous population behaviors are shown to emerge depending on the range of interactions among the individuals. This paper investigates the implications and merits of synchronous behaviors in terms of overall foraging efficiency, where spatial interactions are modelled through a collision avoidance mechanism. Synchronous population behavior, decreasing the chances of collisions, result in more fluid individual trajectories and better foraging returns.

This paper is organized as follows. First, the notion of “foraging swarm” is specified. For the sake of self-containedness, the next section sums up the *SpikeAnts* model, explaining the spiking neurons used and their coupling. Afterwards, the notions of temporal and spatial couplings of agents’ behaviours in a swarm are discussed. Some conjectures on the benefits of synchrony are presented. The experimental setting proposed to study these conjectures is presented. The last section reports on the experimental results, discussing the trajectories of the robotic swarm in a simulated 2D environment. The paper concludes with a discussion and some perspectives for further research.

## Foraging swarm

Basically, the foraging task aims at securing a sufficient amount of food for the (ant) colony at any time. The foraging activity however is itself energy consuming. Therefore it would be inappropriate that all individuals in the colony devote themselves to foraging. Hence the dilemma is: On the one hand, sufficiently many individuals should devote themselves to foraging, but no more; On the other hand, the division of labor between the foraging individuals and the others has to be enforced without any centralized control. Although the division of labor might be resolved by task assignment at the individual level (deciding once for all whether a given individual is a foraging one), such a fixed mechanism would hardly account for the famed flexibility of ant colonies, where the division of labor smoothly adapts to emergencies.

The approach proposed by Liu et al. (2007) involves a probabilistic finite state automaton, where each individual obeys a Markov decision process involving a few states (e.g., resting, foraging, grooming). The probabilities of transitions among states are optimized using evolutionary computation in order to maximize the overall performance of the swarm. The efficiency of this approach thus relies on the size of the swarm, enforcing that the number of individuals in a given state at any point is close to the desired one due to the law of large numbers. Notably, it also requires any individual agent to embed a random generator. Displaying a “truly ran-

dom” decision making process is by no way a basic ability (human beings, for instance, are known to be poor random generators).

## The SpikeAnts model

The proposed individual model is inspired from both Goss and Deneubourg (1988) and Huang and Robinson (1999), where the agent decision results from internal and external factors, and the external factors reflect the other agents’ behavior (social inhibitions).

### Foraging and Social Inhibitions in *SpikeAnts*

Formally, the *SpikeAnts* model involves three states, respectively called *foraging* (active), *sleeping* (inactive) and *observing* (activable), with deterministic transitions (Fig. 1). When in sleeping state, the agent switches to the observing state after some time  $t_s$ ; when in foraging state, the agent switches to the sleeping state after some time  $t_f$ . The agent decision takes place in observing state, taking some cues from the agent’ relatives: essentially, if it sees many other foraging agents (in a sense made precise below) the foraging incentive is low and the agent switches to the sleeping state<sup>2</sup>; otherwise, it switches to the foraging state. This mechanism thus implements social inhibitions, as opposed to e.g. threshold models where the agent decision is based on internal thresholds only (Bonabeau et al., 1998a).

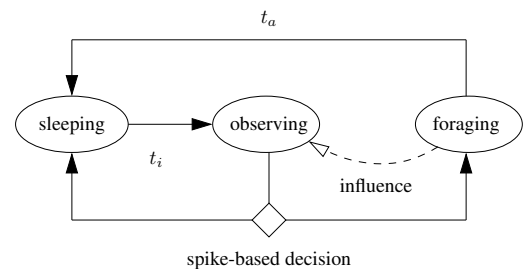


Figure 1: An agent is described by three states and the state transitions are indicated with black arrows. An observing agent decides to forage or not based on local information sent by neighboring active agents (white arrow).

### The competition of two spiking neurons

The agent decision in observing state is made through the competition of two spiking neurons. A model of spiking neuron describes the evolution of an internal variable, the membrane potential; the neuron fires an electrical pulse, called spike, when this membrane potential reaches a given threshold. By connecting spiking neurons to each other and

<sup>2</sup>Note that agents in sleeping state are not necessarily resting but might achieve other tasks as well; the extension of the current model to multi-task settings is a research perspective (see Discussion and Conclusions).

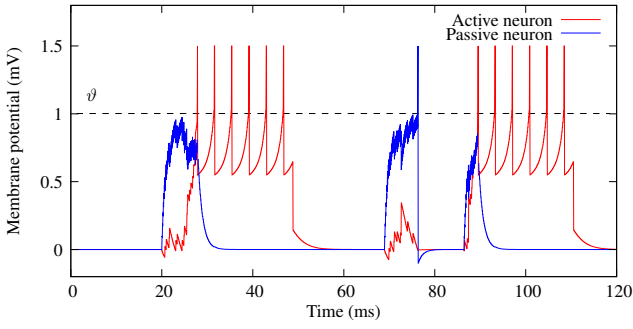


Figure 2: Membrane potentials of active (in dark/red) and passive (in gray/blue) neurons. The dashed line indicates the threshold  $\vartheta$ . Initially sleeping, the agent goes observing at time 20ms. As the active neuron fires before the passive one, the agent goes foraging and the active neuron sends spikes during the whole foraging period, signalling its activity to other agents. It then switches to sleeping state (from circa 50 to 70ms). A second observing period starts thereafter; this time the passive neuron fires before the active one. The agent then switches to sleeping state. At circa 90ms, a third observing period starts, and the agent switches to foraging almost immediately.

having them exchange information through spikes, a rich variety of dynamic activation and synchronization patterns can be obtained.

Formally, an agent is modelled as two spiking neurons, an active one and a passive one. The agent decision (foraging or sleeping) depends on whether the active or the passive neuron fires first. Both neurons are respectively inhibited and activated by the spikes coming from other agents, emitted when these are foraging.

The passive neuron is implemented as a Leaky Integrate-and-Fire (LIF) neuron (Gerstner and Kistler, 2002); the active one is implemented as a Quadratic Integrate-and-Fire (QIF) neuron (Hansel and Mato, 2001); both models have been extensively studied in the literature.

The passive LIF neuron is modelled by a differential equation, which describes the temporal evolution of a potential  $V_p$ . If  $V_p$  exceeds a threshold  $\vartheta$ , the neuron fires a spike and is reset to the resting potential  $V_{\text{reset}}^p$ .

$$\begin{cases} \frac{dV_p}{dt} = -\lambda(V_p(t) - V_{\text{rest}}) + I_{\text{exc}}(t), & \text{if } V_p < \vartheta \\ \text{else fires a spike and } V_p \text{ is set to } V_{\text{reset}}^p \end{cases} \quad (1)$$

where  $\lambda$  is the relaxation constant.  $I_{\text{exc}}(t)$  models instantaneous synaptic interactions. The set of presynaptic neurons is denoted by **Pre**, such that there exists a communication channel from every neuron in **Pre** toward the current neuron. Denoting **Train<sub>i</sub>** the spike train of the  $i^{\text{th}}$  neuron in **Pre**,

$$I_{\text{exc}}(t) = w \sum_{i \in \text{Pre}} \sum_{j \in \text{Train}_i} \delta(t - t_j^i), \quad (2)$$

where  $w$  is a synaptic weight,  $\delta(\cdot)$  is the Dirac distribution and  $t_j^i$  is the firing time of the  $j^{\text{th}}$  spike from the  $i^{\text{th}}$  presynaptic neuron.

The active QIF neuron is described by the evolution of the potential  $V_a$ , compared to the resting potential  $V_{\text{rest}}$  and an internal threshold  $V_{\text{thres}}$ . Additionally, it receives an internal signal  $I_{\text{clock}}$  modelling a gap junction connection:

$$\begin{cases} \frac{dV_a}{dt} = \lambda(V_a(t) - V_{\text{rest}})(V_a(t) - V_{\text{thres}}) \\ \quad + I_{\text{inh}}(t) + I_{\text{clock}}(t), & \text{if } V_a < \vartheta \\ \text{else fires a spike and } V_a \text{ is set to } V_{\text{reset}}^a \end{cases} \quad (3)$$

The choice of this neuron model (Izhikevich, 2007) is motivated by the bistability of the QIF neuron if the reset threshold is greater than the internal threshold ( $V_{\text{reset}}^a \geq V_{\text{thres}}$ ). If  $V_{\text{reset}}^a < V_{\text{thres}}$ , the membrane potential  $V_a$  stabilizes on  $V_{\text{rest}}$  when there is no external perturbation, and the neuron thus exhibits an integrator behavior. When  $V_{\text{reset}}^a \geq V_{\text{thres}}$ , the neuron displays a bursting behavior and fires periodically.

### An Agent Slice of Life

In observing state, the agent decision is thus controlled from the passive LIF neuron (Eq. (1)), the active QIF neuron (Eq. (3)) and an internal clock unit. Both spiking neurons receive the spikes emitted by other neighbor foraging agents (external factors); additionally, the active neuron receives the  $I_{\text{clock}}$  signal emitted by the agent internal clock. The active neuron is activated by the internal signal, and inhibited by the external signals, whereas the passive neuron is only activated by the external signals. Depending on whether the active (respectively the passive) neuron fires first, the agent goes foraging (resp. sleeping).

When the observing agent sees none or few other foraging agents (i.e. receives no or few spike signals from them), the internal signal  $I_{\text{clock}}(t)$  is not counteracted by any external inhibitions and the active neuron fires; it wins the competition and the agent goes foraging (first and third periods in Fig. 2). When in foraging state, the active neuron is bursting and periodically sends spikes to neighbor agents (which process them only if they are in observing state). The agent stays foraging for a time  $t_f$  and then switches to the sleeping state for a time  $t_s$ . This switch is triggered by an internal delay between the clock unit and the active neuron.

If the observing agent perceives many foraging agents, the passive neuron receives many excitatory external signals and it fires first (second period in Fig. 2); the agent switches to sleeping state for a time  $t_s$ .

Overall, the competition between the passive and active neurons thus fully determines the observing agent decision. It is worth noting that the SpikeAnts system is asynchronous<sup>3</sup>. Its temporal dynamics is highly non-linear; in

<sup>3</sup>Differential equations are solved by finite differences, with fixed precision depending on the computational resources.

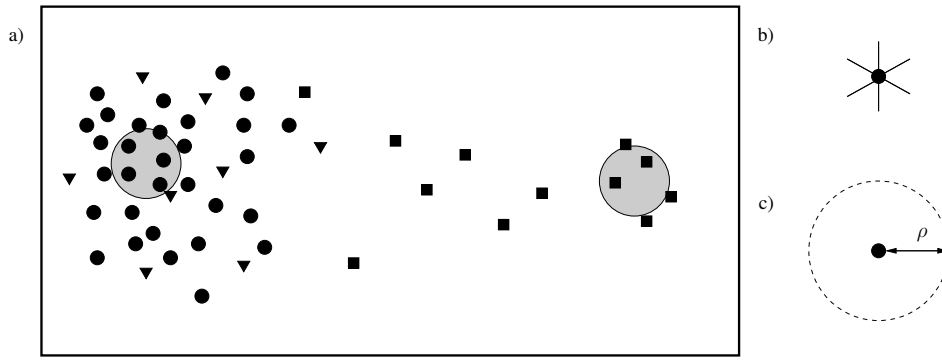


Figure 3: *SpikeAnts* simulation in 2D environment. (a) The arena includes a nest and a food regions (disk); foraging, sleeping and observing agents are respectively indicated with squares, circles and triangles. (b) Proximity zone of an agent. (c) Communication range  $\rho$ .

practice, the number and pace of the spikes received before making a decision vary from one observing period to another. There is no time limit for the observing state; an agent remains in observing state until making a decision.

### Investigating Temporal and Spatial Couplings

In a first study (Chevallier et al., 2010), the temporal couplings induced by *SpikeAnts* have been experimentally studied within an abstract setting, considering each agent as a node in a random graph with a given connectivity rate. Each agent had a fixed position and a fixed, sparse, set of neighbors. In particular, no traveling time from the nest to the food source was accounted for in the foraging activity. Such an abstract setting however does not account for the fact that real and artificial agents alike are moving in a spatial environment, and can hardly be considered as material points.

A more realistic simulated environment is investigated in the present paper. This section introduces the experimental setting and goals.

### Spatial interactions

The study considers a large square 2D arena, the dimension of which is circa 160 times the size of the individual agent. The arena includes the nest, or sleeping place, and the food source, or foraging place (Fig. 3(a)). The region centered on the nest (respectively the source) with radius  $\gamma$  is referred to as nest (resp. source) region. The region centered on the nest with radius  $2\gamma$  is referred to as domestic region ( $\gamma=3\%$  of the arena size in the experiments).

Each agent moves with a constant speed; its communication range  $\rho$  is constant (Fig. 3(c)). The agent is endowed with a set of elementary behaviors:

- In observing state, the agent moves inside the domestic region, with constant speed, except when it sees another agent, where the collision avoidance behavior is executed (see below). Foraging agents within its communication

range send excitatory (resp. inhibitory) signals to its passive (resp. active) neuron.

- Upon the firing of its active neuron, the agent switches to the foraging state for a given time  $t_f$ . Whatever its current position, it goes directly to the food source with constant speed, except when it sees another agent, where the collision avoidance behavior is executed. When arriving in the source region, the agent moves inside this region until the foraging time  $t_f$  is elapsed.
- When switching from foraging to sleeping state, the agent goes directly to the nest region with constant speed, except when the collision avoidance behavior is executed. When arriving in the nest region, the agent moves inside the domestic region until the sleeping time  $t_s$  (starting at the end of the foraging period) is elapsed.
- When switching from observing to sleeping state, the agent stays in the nest region, moving with constant speed, except when the collision avoidance behavior is executed.
- The collision avoidance routine is triggered whenever an obstacle or another agent is located in the proximity zone of the current agent (Fig. 3(b)). The obstacle side is detected as the side (left or right) with higher average sensor activation, and the agent rotates in the opposite direction with a given angle  $\alpha$  ( $\alpha = 5^\circ$  in the experiments). It goes straight ahead in the subsequent time steps (unless some further obstacle enters in its proximity zone, in which case the collision avoidance is triggered anew). When its proximity zone becomes empty again, the agent rotates back to its initial direction.

### Goal of experiments

The experiments are meant to answer two main questions. The first one concerns the temporal couplings between the swarm agents. In the former graph-based setting, several

behavioral regimes were observed depending on the connectivity of the neighborhood graph (agent sociability), ranging from asynchronous behaviors to synchronous aperiodic and synchronous periodic. A 2D environment however involves several sources of variability, which might prevent the swarm from reaching a synchronous behavior. Firstly, the transitions from observing to foraging states are no longer instantaneous as agents must travel from the nest to the source. Secondly, the set of neighbors of each agent varies as the agent moves in observing state, whereas the connectivity was fixed in the previous experiments. Lastly, the agent activity might be perturbed as the collision avoidance routine is executed in priority whenever the agent meets an obstacle or another agent. The question thus is whether the regimes observed in the fixed graph-based setting are still observed in 2D simulations, and whether the transitions from one regime to another depend on the same design parameters.

The second question concerns spatial couplings, and the possible impact of synchronous behavior on the collective foraging efficiency. Whereas synchronous activity is ubiquitous in many living societies and complex systems, the benefits of synchrony remain actively debated. On the one hand, when agents move in a synchronous way as a flock, the chances of collision expectedly decrease and more agents might make it to the source. On the other hand, in asynchronous mode some agents might be deviated from their route to the source due to repeated collision avoidance (“traffic jams”); but it might also be the case that asynchronous agents better share the collective space and the frequency of traffic jams decreases. In order to investigate further the foraging efficiency, two indicators are proposed. The first one counts the *number* of agents arriving at the food source; the second one measures the overall foraging *time*, i.e. the overall number of time units spent by agents in the source region.

## Experimental setting

The experimental setting used to answer the above questions goes as follows. Each agent is simulated as a Khepera robot with eight infra-red sensors and a radio communication module. The communication range  $\rho$  is constant, covering 20% of the arena unless indicated otherwise. Each foraging agent broadcasts its activity signals to all agents with distance less than  $\rho$ ; each observing agent receives the signals of foraging agents on an individual basis. In other words, the simulated setting involves no centralized communication among agents.

At the beginning of each simulation, every agent is sleeping and wakes up after some time, independently and uniformly drawn in  $]0, 2t_s[$ . Each simulation involves 50,000 time steps. All reported results are averaged over 10 independent runs for a given parameter setting. As already mentioned, the *SpikeAnts* model is deterministic; the only source

of variation among simulations comes from the swarm initialization and the uniform agent wake-up times.

Every agent obeys the same *SpikeAnts* model with same parameters as in Chevallier et al. (2010). Foraging and sleeping times are chosen such that their ratio is not an integer, to avoid spurious synchronization effects:  $t_f = 541$  and  $t_s = 457$  time steps. Spiking neurons are simulated using a clock-driven simulator and Runge-Kutta method for differential equation approximations with a small time step of 0.1ms to achieve numerical stability.

## Experimental Results

This section reports on the temporal and spatial couplings observed in the 2D simulation of *SpikeAnts*.

### Emergence of Temporal Self-Organization

The temporal coupling at the population level is displayed in Fig. 4, reporting the number of active agents vs the simulated time step. Three behavioral regimes emerge depending on the parameter setting. An *asynchronous* regime (Fig. 4, (A)) is observed for low communication ranges; agents individually and asynchronously decide to go foraging, with an average number of 30 foraging agents in each time step out of 100 agents. Another *synchronous aperiodic* regime sees the emergence of sub-populations of agents, that synchronously decide to go foraging; still the size of the foraging subpopulation varies from one period to another one, and the foraging subpopulations with same size gather distinct agents in each period (Fig. 4, (B)). Finally, the *synchronous periodic* regime involves a few persistent subpopulations (two in Fig. 4, (C)), which alternate and go foraging. The agent trajectories in all three regimes are shown in Fig. 5, displaying different spatial patterns.

The emergence of these regimes has been explained from a few *SpikeAnts* design parameters (Chevallier et al., 2010). The first factor is the communication range  $\rho$ , given as percentage of the arena covered when agents broadcast/receive the foraging signal. For a low  $\rho$ , the agent decision is based on a few local cues; for a high  $\rho$ , every agent can reliably estimate the number of currently foraging agents. The second factor is called *receptivity* and characterizes the strength of interactions between agents; it is expressed as the ratio between the connection weight  $w$  and the sub-threshold range (depending of the resting potential  $V_{rest}$  and the firing threshold  $\vartheta$ ):  $\left| \frac{w}{\vartheta - V_{rest}} \right|$ . With a high interaction strengths, a few spikes can trigger the agent decision; small variations in the received information lead to different decisions. With a low interaction strength, agent decisions are based on many signals; the number of spikes needed to reach the threshold is high and the agent decision thus is more stable.

The transition between all three regimes is made precise using an entropy indicator defined as follows. Let  $n_t$  denote the number of foraging agents at (simulated) time  $t$ ,

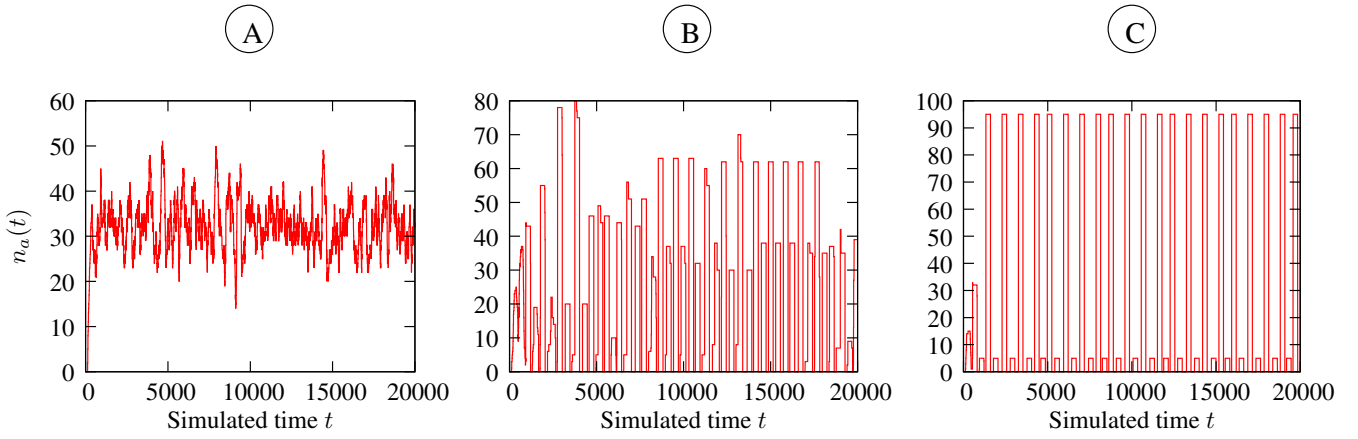


Figure 4: Three behavioral regimes emerge in the population: (A) Asynchronous, (B) Synchronous aperiodic and (C) Synchronous periodic. Each graph reports the number of active agents *vs* the simulated time step.

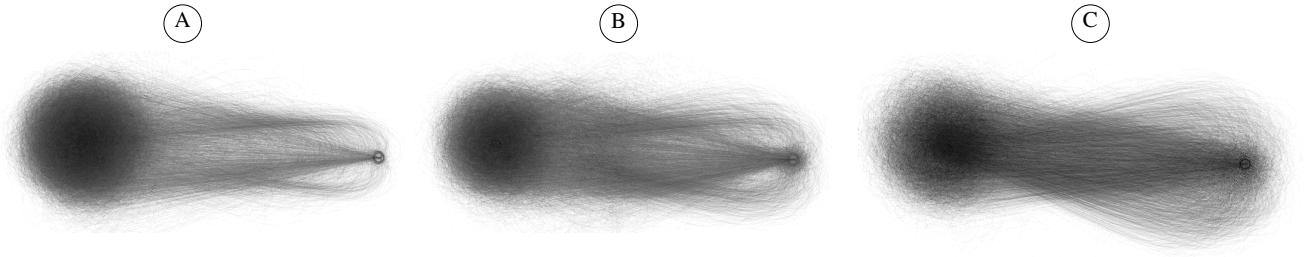


Figure 5: Trajectories followed by agents during a representative run, respectively in asynchronous (A), synchronous aperiodic (B) and synchronous periodic (C) regimes. Darker lines indicated the most often visited paths.

with  $0 \leq n_t \leq M$  and  $M = 100$  is the overall number of agents. Let  $p_n$  denote the percentage of time steps such that  $n_t = n$ . The foraging entropy is classically defined as  $H = -\sum_{n=0}^M p_n \log p_n$ . The phase diagram, reporting the entropy *vs* the two control parameters of agent sociability and receptivity, is displayed in Fig. 6.

The asynchronous regime is characterized by a medium entropy value (circa 3) as the  $n_t$  values are tightly distributed around a mean value. This regime emerges in populations with low communication range and high interaction strength. For a medium communication range and weak interactions, a synchronous aperiodic activity is observed, with high entropy ( $H \approx 4$ ) as the sub-population sizes vary from 10 to 80 agents. A stable synchronous periodic regime, characterized by a low entropy value ( $H \approx 1$  since the sub-population sizes are very stable), is observed for a large communication range and strong interactions. On Fig. 6, the synchronous periodic regime emerges for a communication radius which cover nearly all the arena ( $\rho = 80\%$ ). Complementary experiments show that for a smaller  $t_f/t_s$  ratio the transition from asynchronous to synchronous regimes is shifted on the left, and occur for smaller communication range ( $\rho = 30\%$ , results omitted for brevity).

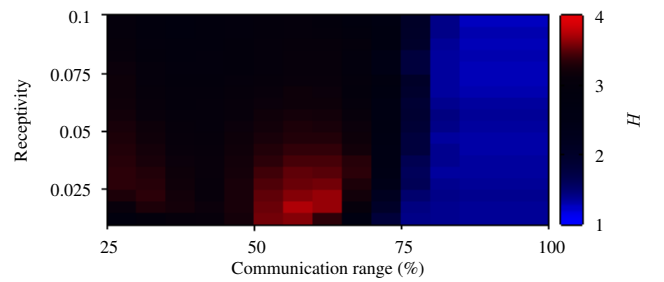


Figure 6: Phase diagram of the temporal coupling: foraging entropy *vs* the agent sociability (*x* axis) and receptivity (*y* axis).

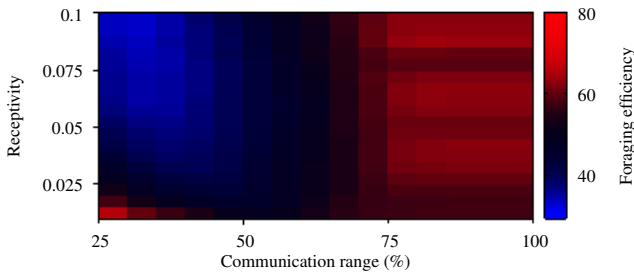


Figure 7: Foraging efficiency defined as the percentage of foraging agents that arrive in the source region.

### Mixed Benefits of Synchronous Foraging

As mentioned earlier on, it was expected that synchronous regimes would benefit the foraging activity through decreasing the chance of collisions. The foraging efficiency measuring the fraction of foraging agents making it to the source is displayed in Fig. 7; on average, 80% of the foraging agents arrive in the source region in synchronous periodic regime, as opposed to less than 40% in asynchronous regime. The lower foraging efficiency in asynchronous regime is related to the “price of anarchy”: more chances of collisions slow down the agents on their way to the source region, to the point that the foraging period ends up for most agents before they even reach the source, and they go back to the nest with empty hands.

Additional experiments are conducted to examine the sensitivity of the synchrony benefits when increasing the foraging time  $t_f$ . For larger  $t_f$  values, all agents eventually arrive at the source sooner or later. It thus comes naturally to consider the agent traveling time  $t_r$ . By construction,  $t_{min} \leq t_r \leq t_{max} = t_f$ , where  $t_{min}$  is the minimum time needed to go from the nest to the source. Let us accordingly define the *foraging loss* as the excess time wasted in the travel from the nest to the source,  $L = \frac{t_r - t_{min}}{t_{max} - t_{min}}$ . A contrasted picture then appears (Fig. 8): the foraging loss is minimum in asynchronous regime (less than 50%), and it increases when the swarm switches to synchronous aperiodic or periodic regimes (up to 65%). Agent paths shown on Fig. 5 corroborate these results: agents in synchronous regimes display more spatially distributed trajectories than in asynchronous regime, thus increasing the traveling time.

This experiment suggests that synchronous foraging entails opposite effects: while less agents arrive at the food source in asynchronous regime, the overall foraging time remains higher than for synchronous regimes. Additional experiments will examine these mixed effects in more depth.

### Discussion and Perspectives

This paper has presented the distributed, decentralized and deterministic swarm model *SpikeAnts*, accounting for the emergence of synchronous behaviors and division of labor in

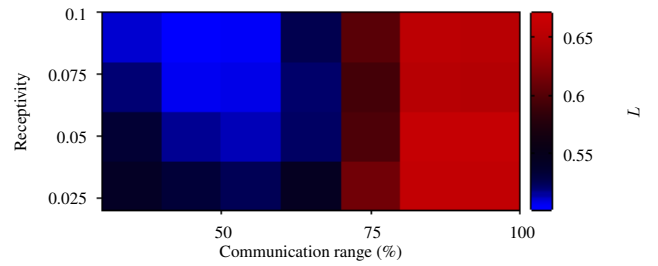


Figure 8: Foraging loss  $L = \frac{t_r - t_{min}}{t_{max} - t_{min}}$ , measuring the foraging time wasted in the travel from the nest to the source.

a foraging swarm. Depending on the communication range and interaction strength among agents, the swarm behavior ranges from an asynchronous regime, where every agent independently makes its decisions, to a synchronous periodic regime where two persistent sub-populations alternate and go foraging.

A most interesting and unexpected experimental result concerns the mixed effects of synchrony. Quite a few authors have advocated the benefits of synchrony for division of labor: temporal coactivation of individuals enhance the information exchange and the cohesion of the population (Robinson, 1992; Bonabeau et al., 1998b; Delgado and Solé, 2000); synchrony also provides an intrinsic mechanism of mutual exclusion (Hatcher et al., 1992). In a 2D framework however, agent synchronization entails some spatial couplings through the collision avoidance mechanism. The experimental evidence suggests that synchronous flocking behaviors decrease the chances of collision (and more agents arrive at the target destination), but increase the traveling time (and agents have less time to achieve the task when arrived at destination).

Additional experiments will be needed to investigate these effects, and a first perspective is to implement *SpikeAnts* on a physical robotic platform. As already mentioned, *SpikeAnts* was designed to comply with limited memory and computational resources. Along the same lines, *SpikeAnts* will be extended to deal with several tasks of diverse priorities (e.g., collecting energy and rescuing the swarm robots out of energy). The question is whether and when the swarm will demonstrate several sub-populations attending the different tasks in a synchronous way, and how the division of labor may take place depending on the experimental setting.

A further question regards how the collective regime will be modified under external perturbations in the environment, and how the swarm adapts its response. A yet further stage will be to consider autonomous and adaptive agents, e.g. controlling their foraging time or interaction strength depending on their internal state and individual agenda.



## Acknowledgments

This work was made possible by the European Union FET Proactive Initiative: Pervasive Adaptation funding the Symbion project under grant agreement 216342.

## References

- Bayindir, L. and Sahin, E. (2007). A review of studies in swarm robotics. *Turkish Journal of Electrical Engineering*, 15(2).
- Beshers, S. and Fewell, J. (2001). Models of division of labor in social insects. *Annual Review of Entomology*, 46(1):413–440.
- Bonabeau, E., Theraulaz, G., and Deneubourg, J. (1998a). Fixed response thresholds and the regulation of division of labor in insect societies. *Bulletin of Mathematical Biology*, 60(4):753–807–807.
- Bonabeau, E., Theraulaz, G., and Deneubourg, J. (1998b). The synchronization of recruitment-based activities in ants. *BioSystems*, 45:195–211.
- Branham, M. and Greenfield, M. (1996). Flashing males win mate success. *Nature*, 381(6585):745–746.
- Camazine, S., Deneubourg, J. L., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2001). *Self-Organization in Biological Systems*. Princeton University Press.
- Chevallier, S., Paugam-Moisy, H., and Sebag, M. (2010). Spikeants, a spiking neuron network modelling the emergence of organization in a complex system. In *NIPS*, pages 379–387.
- Cole, B. (1991). Short-term activity cycles in ants: Generation of periodicity by worker interaction. *The American Naturalist*, 137(2).
- Delgado, J. and Solé, R. (2000). Self-synchronization and task fulfilment in ants colonies. *J. Theoretical Biology*, 205:433–441.
- Dorigo, M., Tuci, E., Groß, R., Trianni, V., Labella, T., Nouyan, S., Ampatzis, C., Deneubourg, J.-L., Baldassarre, G., Nolfi, S., Mondada, F., Floreano, D., and Gambardella, L. (2005). The SWARM-BOTS project. In *Proc. of Workshop on Swarm Robotics*, volume 3342 of *LNCS*, pages 31–44. Springer.
- Gerstner, W. and Kistler, W. (2002). *Spiking Neuron Models: Single Neurons, Population, Plasticity*. Cambridge University Press.
- Goss, S. and Deneubourg, J. (1988). Autocatalysis as a source of synchronised rhythmic activity in social insects. *Insectes Sociaux*, 35(3):310–315.
- Hansel, D. and Mato, G. (2001). Existence and stability of persistent states in large neuronal networks. *Physical Review Letters*, 86(18):4175–4178.
- Hartbauer, M. and Romer, H. (2007). A novel distributed swarm control strategy based on coupled signal oscillators. *Bioinspiration and Biomimetics*, 2(3):42–56.
- Hatcher, M., Tofts, C., and Franks, N. (1992). Mutual exclusion as a mechanism for information exchange within the ant nests. *Naturwissenschaften*, 79:32–34.
- Hauert, S., Winkler, L., Zufferey, J.-C., and Floreano, D. (2008). Ant-based swarming with positionless micro air vehicles for communication relay. *Swarm Intelligence*, 2(2):167–188.
- Huang, Z. and Robinson, G. (1999). *Social control of division of labor in honey bee colonies*, pages 165–186. Birkhäuser.
- Izhikevich, E. (2007). *Dynamical systems in neuroscience: the geometry of excitability and bursting*, chapter One-Dimensional Systems. MIT Press.
- Krafft, B. and Pasquet, A. (1991). Synchronized and rhythmic activity during the prey capture in the social spider. *Insectes Sociaux*, 38(1):83–90.
- Labella, T., Dorigo, M., and Deneubourg, J.-L. (2006). Division of labor in a group of robots inspired by ants’ foraging behavior. *ACM Trans. Auton. Adapt. Syst.*, 1(1):4–25.
- Leinaas, H. (1983). Synchronized moulting controlled by communication in Group-Living collembola. *Science*, 219(4581):193–195.
- Liu, W., Winfield, A., Sa, J., Chen, J., and Dou, L. (2007). Towards energy optimisation: Emergent task allocation in a swarm of foraging robots. *Adaptive Behavior*, 15(3):289–305.
- Panait, L. and Luke, S. (2004). A pheromone-based utility model for collaborative foraging. In *Proc. of AAMAS*, pages 36–43.
- Parker, L. E. (2008). Distributed Intelligence: Overview of the Field and its Application in Multi-Robot Systems. *Journal of Physical Agents*, 2(1).
- Paugam-Moisy, H. and Bohte, S. (2011). *Handbook of Natural Computing*, chapter 10. Computing with Spiking Neuron Networks. Springer. (in press).
- Robinson, G. (1992). Regulation of division of labor in insect societies. *Annual Review of Entomology*, 37:637–665.
- Trianni, V. and Nolfi, S. (2009). Self-organizing sync in a robotic swarm: a dynamical system view. *Transactions on Evolutionary Computation*, 13:722–741.
- Wischmann, S., Hlase, M., Knabe, J. F., and Pasemann, F. (2006). Synchronization of internal neural rhythms in multi-robotic systems. *Adaptive Behavior*, 14:117–127.

# Degeneracy Enriches Artificial Chemistry Binding Systems

Ed Clark<sup>1</sup>, Adam Nellis<sup>1</sup>, Simon Hickinbotham<sup>1</sup>, Susan Stepney<sup>1</sup>, Tim Clarke<sup>1</sup>, Mungo Pay<sup>1</sup>, Peter Young<sup>1</sup>

<sup>1</sup>York Centre for Complex Systems Analysis, University of York, UK, YO10 5GE

edclark@cs.york.ac.uk

## Abstract

We hypothesise that degeneracy in the components of an artificial chemistry (AChem) facilitates the complexity of the system as a whole. We introduce definitions of degeneracy and redundancy, and show how these quantities can be calculated for the binding system of an AChem.

We present a case study using the AChem Stringmol, in order to support our hypothesis. We demonstrate that the binding system in Stringmol has degeneracy and we create a deliberately poor variant: ‘sticky-Stringmol’, that has a binding system with no degeneracy. Comparing sticky-Stringmol to Stringmol, we note the loss of many simulation artifacts that have been used as evidence of the complexity of Stringmol, including: emergent macro-mutations, hypercycles, sweeps and parasite evasion. These results are evidence that degeneracy in the components of an AChem facilitates the complexity of the system as a whole.

## Introduction

Degeneracy is the ability of elements that are different, in some respect, to perform the same role in some, but not all, situations. Degeneracy is a noticeable property of many biological systems, and is observable on many scales within those systems [7] and has been linked to the evolvability and robustness of these systems [16]. Examples range from molecular interactions and gene networks [7], the connectivity of neurons in the brain [13], through to social networks [15]. Complexity and degeneracy have been strongly linked [14]. Attempts have been made to describe these concepts into mathematically meaningful, and consequently unambiguous formulae [14][7].

Just as degeneracy can be observed on many scales in nature, so it should be in artificial chemistries (AChems) that aspire to achieve the levels of complexity that exist in the natural world. We hypothesise that degeneracy in the components of an AChem will facilitate complexity of the system as a whole. We introduce measures of degeneracy and redundancy in terms of an ‘interaction function’ between two sets. We use binding between two sets of chemicals in an AChem (defined below) as a concrete example of an interaction function. We demonstrate that the degeneracy in

the binding system of Stringmol [10] is particularly important for the complexity of the AChem as a whole.

When presenting the complexity of an AChem, it is standard practice to present simulation results and focus on an artifact that the system has been able to produce as evidence of the complexity of the AChem. Examples of artifacts include: the ability to ‘compute’ prime numbers [1]; the generation of cooperative organisations [9], hypercycles [10] and autocatalytic sets [12]. However, the complexity available in current AChems is still well below that of the natural world.

The presentation of simulation artifacts is currently the only available way to evaluate AChems (see [4]). As such, two chemistries that produce different types of artifact can only be compared in a qualitative manner. Progress has been made on formalising artifacts in chemistries, and automating the discovery of autocatalytic sets [12] and organisations [5]. However, simulation artifacts can only be measured *a posteriori*: they can not be determined at design stage. The degeneracy measure we introduce can be applied at the design stage to the components of an AChem, thus allowing sources of complexity to be designed in.

## Binding in AChems

In this paper, we focus on degeneracy in the context of binding in AChems. In the ‘ $(S, R, A)$ ’ definition of AChems [4],  $S$  is a set of chemicals,  $R$  is a set of reactions between the chemicals and  $A$  is the algorithm that applies reactions from  $R$  to chemicals from  $S$ . For example, if the chemicals in set  $S$  are integers, then the set  $R$  of reactions might contain all reactions of the form:

$$a + b \mapsto c \quad \text{if } c = \frac{a}{b} \text{ is an integer.} \quad (1)$$

This is the prime number generation chemistry [1].

The important point for this discussion is the binding rule: “if  $\frac{a}{b}$  is an integer”. This can be viewed as an “if-then” statement: if the binding rule is true, the reaction may proceed. The left hand side (LHS) of the reaction, “ $a + b$ ”, is the if part of this statement. The right hand side (RHS) of the reaction, “ $\mapsto c$ ”, is the then part. Looking at the chemistries reviewed in [4], the vast majority have a trivial LHS, where

the *if* simply tests if two molecules are presented by the algorithm, *A*. The only AChems we are aware of with a non-trivial LHS are Primes [1], AlChem [9], Stringmol [10], Molecular Classifier Systems [3] and RBN-world [8].

AlChem (level 0) had relatively simple binding, which resulted in the collapse of the system into ‘self-replicators’. In AlChem (level 1), binds that would result in reactions that propagate self replicators were restricted. As a result of the enriched binding rule, AlChem (level 1) produced more complex artifacts, including ‘cooperative organisations’ [9]. This example helps support our hypothesis that binding is an important component of an AChem, and that changes to this component can change the level of complexity observed in the system.

## Organisation of the Paper

We define degeneracy and redundancy in an unambiguous manner, and introduce methods to measure these quantities. We justify introducing a new measure of degeneracy instead of adopting previously published measures. We use our measures to analyse the binding system used in Stringmol and demonstrate that the binding system is capable of producing degeneracy. We also use these measures to demonstrate that ubiquitous binding is unable to produce degeneracy. We use ubiquitous binding to define a deliberately poor Stringmol variant: ‘sticky-Stringmol’. We replicate the experimental procedures of [10] in order to compare the artifacts of ‘sticky-Stringmol’ and Stringmol. We give an overview of the previously undetected phenomena of ‘parasite evasion’ in Stringmol containers. The two mechanisms by which the container is able to survive a potentially fatal parasite are linked to binding. We also find that sticky-Stringmol containers are unable to evade a parasite.

## Degeneracy and Redundancy

We formally introduce and define redundancy and degeneracy in abstract terms, and provide a worked example calculating the redundancy and degeneracy of the binding system of a fictitious AChem.

In order to make an unambiguous statement of redundancy or degeneracy, one must state three pieces of information: Two sets of elements, *A* and *B*, that are being compared, and the method of comparison, defined by an ‘interaction function’,  $f : A \times B \mapsto \{0, 1\}$ , stating whether an element of *A* and an element of *B* interact or not.

If we consider an arbitrary element,  $a_m$  of set *A*, we can define a subset  $B_{a_m} \subseteq B$ , in terms of  $(a_m, f, B)$ , containing all the elements of *B* that  $a_m$  interacts with:

$$B_{a_m} = \{b \in B \mid f(a_m, b) = 1\}. \quad (2)$$

Elements  $a_m$  and  $a_n$ , are *redundant* if

$$\mathcal{R}(a_m, a_n \mid f, B) \Leftrightarrow B_{a_m} = B_{a_n}. \quad (3)$$

Elements  $a_m$  and  $a_n$ , are *degenerate* if

$$\mathcal{D}(a_m, a_n \mid f, B) \Leftrightarrow (B_{a_m} \neq B_{a_n} \ \& \ B_{a_m} \cap B_{a_n} \neq \emptyset). \quad (4)$$

The definitions in equations 3 and 4 equip us to deal with questions concerning individual examples such as ‘are  $a_1$  and  $a_2$  degenerate or redundant in a given context’. The ability to determine if  $\mathcal{D}(a_1, a_2 \mid f, B)$  is true, does not equip us to answer more general questions, such as what is the degeneracy of a set in a given context,  $\mathcal{D}(A \mid f, B)$ .

Degeneracy and redundancy, even when clearly defined between elements, have a non-trivial interaction within a set. Consider: sets  $C, D = \{a, b, c, d, f, g\}$ , and some interaction function *f* that causes the resulting matrix, which can be viewed as a network, to contain examples of both degeneracy and redundancy, see figure 1a. Consider also  $A, B = \{a, b, c, d, e, f, g\}$ , where *e* is part of a redundant set with *a*, see figure 1b. True measures of degeneracy and redundancy should detect that the redundancy of the set *C* is different from the set *A*. However, is the degeneracy of set *C* the same as the degeneracy of set *A*? If one wishes to maintain degeneracy of a set and redundancy of the set as orthogonal concepts, then the answer to this question must be ‘no’. If one answers ‘yes’, then the concept of the degeneracy of a set becomes conflated with the redundancy of the set. As a result of this conflation, such measures of degeneracy lose their value, as the results they give may be skewed by redundancy. This is why we introduce a new measure of degeneracy, rather than adopting an existing measure. The key to understanding the relationship between degeneracy and redundancy, is knowing that it is possible to measure the redundancy of a set without regard for the degeneracy of a set, but not the other way around.

It is, however, possible to construct a measure of degeneracy of a set that does not suffer from this conflation with redundancy, keeping the mathematical concepts of degeneracy and redundancy of sets orthogonal. We introduce such a measure here. Our method avoids the conflation problem by accounting for the redundancy of the two sets (in the contexts of an interaction function *f*) and constructing new sets that have no redundancy. The set  $\hat{A}$  is constructed from the set *A* (in the context of set *B* and the interaction function *f*) such that the elements  $\hat{a} \in \hat{A}$  are the redundant sets of *A*. We can construct  $\hat{B}$  in a similar manner. Note that it makes no difference if we construct  $\hat{B}$  in the context of  $\hat{A}$  or the context of *A*. These constructions can take place in any order and all examples of degeneracy that exist in  $(A, B)$  are maintained in  $(\hat{A}, \hat{B})$ .

Each element  $\hat{a}$  of the reduced set  $\hat{A}$  is itself a set containing one or more redundant elements from *A*. It is on these redundant sets that we base our measure of degeneracy. If we reconsider the above thought experiment, it can be seen that the element *e* will join an existing redundant set, see figure 1 parts (c) and (d). Consequently it will not

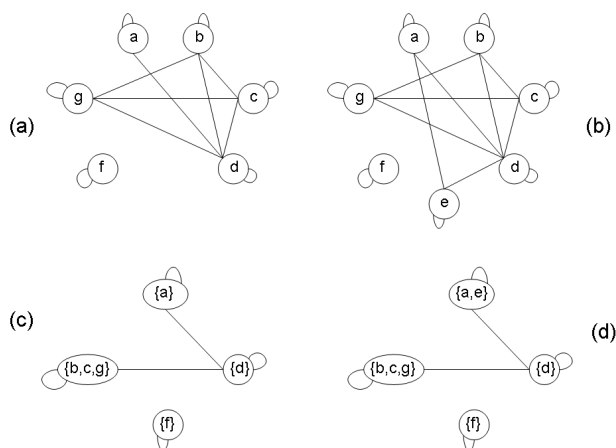


Figure 1: A network example of interaction between elements of a set. Two nodes  $x$  and  $y$  are joined with an edge if  $f(x, y) = 1$ . (a) is an example of an interaction containing examples of both degeneracy and redundancy by the definitions given in equations 3 and 4. In (b) the element ‘ $e$ ’ has been added. The elements  $a$  and  $e$  form a redundant set  $\{a, e\}$  as they both bind to elements  $\{a, e, d\}$ . (c) and (d), show the same relationships as (a) and (b) respectively, but in terms of redundant sets rather than elements.

affect a measure of degeneracy that is based on the elements of  $\hat{A}$  (the redundant sets of the elements of  $A$ ), instead of the elements of  $A$ .

We follow the definitions of redundancy and degeneracy for pairs of elements, and define redundancy and degeneracy for sets. Firstly, we consider an arbitrary element of set  $\hat{A}$ ,  $\hat{a}$ , and define the subset  $\hat{B}_{\hat{a}_m} \subseteq \hat{B}$ . This contains all the redundant sets of  $\hat{B}$  that  $\hat{a}$  interacts with:

$$\hat{B}_{\hat{a}_m} = \{\hat{b} \in \hat{B} \mid f(\hat{a}_m, \hat{b}) = 1\}. \quad (5)$$

We define the *redundancy of the set  $A$* , in the context of  $(f, B)$ , as the set of sizes of redundant sets of  $A$ :

$$\mathcal{R}(A \mid f, B) = \{|\hat{a}| \mid \hat{a} \in \hat{A}\}. \quad (6)$$

$\mathcal{R}(A \mid f, B)$  takes the form of a set of size  $|\hat{A}|$ ; the elements of this set are the sizes of the sets  $\hat{a} \in \hat{A}$ .

We define the *degeneracy of the set  $A$*  in the context of  $(f, B)$ :

$$\mathcal{D}(A \mid f, B) = \{|\hat{B}_{\hat{a}}| \mid \hat{a} \in \hat{A}\}. \quad (7)$$

$\mathcal{D}(A \mid f, B)$  also takes the form of a set of size  $|\hat{A}|$ ; the elements of this set are the numbers of redundant sets in  $\hat{B}$ , that each element  $\hat{a} \in \hat{A}$  interacts with.

### Worked Example

We define  $A, B = \{a, b, c, d, e, f, g\}$  to be all the chemicals in our fictitious chemistry. Note that for the purposes of this

example, we do not need to specify the reaction rule, as the products of reactions do not concern us in this calculation. We assume the binding rule returns a probability; we can apply a threshold at zero in order to construct an interaction function  $f$ . The result of the thresholding is shown in figure 2a. As it contains only binary values, it is an interaction matrix and the definitions of degeneracy and redundancy given in equations 3 and 4 apply (as in figure 1b).

We now construct the redundant sets: In figure 2a the row  $a$  and the row  $e$  have the same values, as such they are redundant under the definition given in equation 3. Similarly, rows  $b, c$  and  $g$  all have the same values. We can construct the redundant sets  $\hat{A} = \{\{a, e\}, \{b, c, g\}, \{d\}, \{f\}\}$ ; if we apply the same process to the columns, we obtain the reduced matrix shown in figure 2b (as in figure 1d).

The sizes of the redundant sets, shown in the row labels in figure 2b, make up the redundancy set,  $\mathcal{R}(A \mid f, B) = \{2, 3, 1, 1\}$ , shown in figure 2c. In order to quantitatively compare binding systems from different chemistries of different sizes we scale the redundancy set by dividing the values in the set by the average redundancy. The average redundancy is given by the sum of the set sizes divided by the number of sets; in this case  $(2+3+1+1)/4 = 7/4$ . The scaled redundancy set is the redundancy set divided by the average redundancy, shown in figure 2c.

From the reduced interaction matrix in figure 2b, it is also possible to calculate the degeneracy set. The degeneracy of set  $A$  is obtained by summing the respective rows in the reduced interaction matrix in figure 2b, the result,  $\mathcal{D}(A \mid f, B) = \{2, 2, 3, 1\}$ , is shown in figure 2d. The degeneracy of set  $B$  would be obtained by summing the columns. Note that the calculation of degeneracy is not based on the elements of set  $A$ , but is instead based on  $\hat{A}$ , the redundant sets of  $A$ .

We rescale the degeneracy set by dividing the degeneracy set by the average degeneracy. The average degeneracy is calculated by summing all the elements in the interaction matrix in figure 2b and dividing that by the number of rows,  $|\hat{A}|$ . In this case the average degeneracy is  $8/4=2$ . The scaled degeneracy set is shown in figure 2d.

These scaled sets can be used to compare the spread of redundancy and degeneracy when the systems being compared are of different sizes. A scatter plot is ideal for such a comparison, the rescaled degeneracy and redundancy sets from the worked example are shown in figure 3. If the systems being compared are the same size then it is appropriate to use the unscaled sets, allowing comparison of both the relative spread and the actual values of redundancy and degeneracy.

## Results

We apply our measures of degeneracy and redundancy (defined in equations 6 and 7) to the binding system used in Stringmol and to ‘ubiquitous binding’ (all molecules bind). This makes ubiquitous binding a good candidate for testing

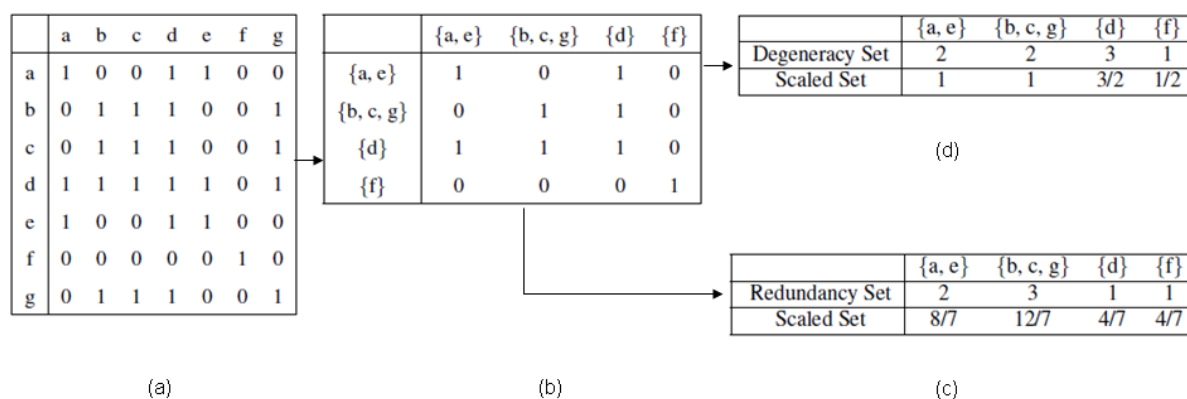


Figure 2: (a): The interaction matrix:  $\{a, b, c, d, e, f, g\}$  are the chemicals of set  $A$ , 1 indicating two molecules bind and 0 indicating two molecules do not bind. (b), the reduced matrix: The elements of  $\hat{A}$  are the redundant sets of  $A$ , these sets are given explicitly as the row and column labels. (c): The redundancy set and scaled redundancy set, for set  $A$ . The values of the redundancy set are the number of elements in row labels of (b). The scaled redundancy set is obtained by dividing the unscaled set by the average redundancy. (d): The degeneracy set and scaled degeneracy set for set  $A$ . The values of the degeneracy set are the number of ones on each row in the reduced matrix (b). The scaled degeneracy set is obtained by dividing through by the average degeneracy in  $\hat{A}$ .

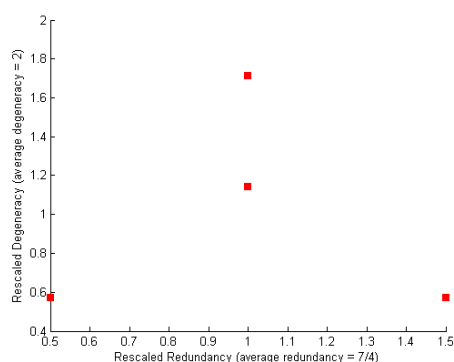


Figure 3: Redundancy and degeneracy scatter plot for set  $A$  in the worked example. Each point in the scatter represents a redundant set (an element of  $\hat{A}$ ). The position on the redundancy axis is given by the scaled redundancy set shown in figure 2(c). Similarly, the position of the degeneracy axis is given by the scaled degeneracy set shown in figure 2(d). The redundancy and degeneracy sets are scaled such that the center of mass of the scatter plot is at (1, 1).

our hypothesis: that a more simplistic approach to binding will negatively impact the complexity of the artifacts observed in simulations. We apply the methodology of [10] to sticky-Stringmol and compare our results. We then describe how mid run parasites are evaded in Stringmol and how the mechanism for evasion is lost in sticky-Stringmol.

In our general comparison with Stringmol, as well as in the parasite trial, sticky-Stringmol is effectively the control experiment. By having a deliberately poor variant of String-

mol, we are able to establish which simulation artifacts are dependent on the degeneracy of the binding rule.

### Measuring degeneracy

The Stringmol alphabet is 33 characters: 7 ‘functional’ characters  $\{\$, \>, \wedge, \text{?}, =, \%, \}$ , and 26 ‘non-functional’ characters  $\{A - Z\}$ . Functional characters in Stringmol can contribute towards a bind site, but they contribute half as much as non-functional characters (for the full details, see [11]). We present results for the reduced character set:  $\{AB\%CD\}$ , containing 1 functional character and 4 non-functional characters, as the calculation for the full character set is intractable. We use the tailored Smith-Waterman algorithm [11] to calculate the bind strength of all strings of length 6 from this alphabet, and threshold at a Smith-Waterman score of 0.75 to produce an interaction function.

Degeneracy and redundancy for the set of all strings of length 6 are shown in figure 4. We also present the degeneracy and redundancy for the ubiquitous binding system used in sticky-Stringmol (under the same conditions) on the same figure, to allow a direct comparison of the properties of the two binding systems. We argue only that the tailored Smith-Waterman algorithm is capable of producing degeneracy and redundancy, not the specific levels of this which can be achieved for a string of arbitrary length. For ubiquitous binding, the matrix of interactions is filled with ‘1’ in every element, the result that it is maximally redundant scales to strings of any length and character set.

Ubiquitous binding has trivial redundancy and no degeneracy, which makes the comparison in figure 4 appear unnecessary. However, this is a simple example of a general technique that, for a given alphabet, can be used to compare

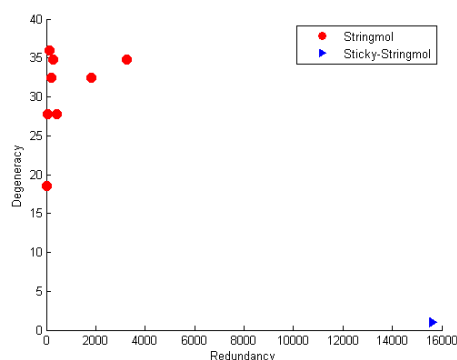


Figure 4: Comparison of the properties of the binding systems of Stringmol and sticky-Stringmol. The comparison shows the unscaled redundancy and degeneracy sets as the two systems contain the same number of elements. Each of the circles is associated with a redundant set. All of the circles together represent the Stringmol binding system. Sticky-Stringmol has only one triangle, as all of its elements form a single redundant set. It can be seen that the Stringmol binding system has a spread of both degeneracy and redundancy, whereas sticky-Stringmol’s ubiquitous binding has only trivial redundancy.

two or more binding systems of any level of degeneracy and redundancy.

### Degeneracy Affecting Simulation Artifacts

Stringmol is an AChem that encodes ‘microprograms’ as strings of characters. We give a brief overview of Stringmol here (for more details, see [11]). Each molecule is a string of characters that encodes a sequence of instructions, making use of pointer manipulations. A number of molecules are initialised in a reaction container. Pairs of molecules in the container are given an opportunity to react by the physics engine. In a biological system, although there may be thousands of different species of molecules, we note that in the majority of possible pairwise combinations, the number of molecules that each molecule interacts with is relatively small. As a result, care was taken in the design of Stringmol to check if two molecules could bind or not via a rich binding system. We made use of a tailored variant of the Smith-Waterman string-matching algorithm [11]. The Smith-Waterman algorithm is used in the study of Biology to compare the similarity of two sequences of DNA. In Stringmol, the Smith-Waterman based binding algorithm determines:

- with what probability two molecules bind;
- given that they bind, how the molecules are aligned;
- which molecule is the executing microprogram, and where its pointers are initialised.

Simulation Artifacts in Stringmol	Binding system property	
	Degeneracy	No Degeneracy
Self replication	✓	✓
Parasites	✓	✓
Random walks	✓	✓
Sweeps	✓	×
Macro-mutations	✓	×
Hypercycles	✓	×
Parasite evasion*	✓	×

Table 1: Comparison of system level properties, used to evaluate the level of the complexity of an AChem. Degeneracy denotes the original Stringmol binding system, No Degeneracy denotes the ubiquitous binding system used in the sticky-Stringmol variant. \* ‘Parasite evasion’ was not originally on the list of Stringmol’s properties published in [10]; we introduce it and present evidence that it occurs in Stringmol, but not in sticky-Stringmol.

In 1000 trials of Stringmol, numerous phenomena were observed, including the emergence of hypercycles (two mutually dependent molecules), macro-mutations (non-point based mutations), sweeps (change of dominant replicase, other than by a random walk) and parasites [10].

The hypothesis is that: by changing the level of degeneracy in the binding system of an AChem, we will alter the simulation artifacts. We investigate this proposed link by comparing Stringmol and sticky-Stringmol (with the full character set). The degeneracy and redundancy of these two binding systems (for a particular character set) is shown in figure 4. We repeated the experimental protocol of [10], running 500 trials of sticky-Stringmol to observe the diversity that arises from a mono-culture. Table 1 shows a comparison of the observed simulation artefacts. Sticky-Stringmol makes use of ubiquitous binding (no degeneracy), as opposed to the Smith-Waterman based binding of Stringmol (degeneracy).

The instruction set used in sticky-Stringmol is the same as the instruction used set in Stringmol. This might lead one to expect they should have computational artifacts of equal complexity; we find this is not the case, see table 1. These results show that a naive binding system, such as ubiquitous binding, can suppress complexity in an AChem. This identifies binding systems to potentially be an important aspect in all AChems.

These results indicate that the binding system has a strong effect of the overall complexity of the system.

### Parasite Evasion in Stringmol Containers

Having presented the main results of the paper, we now present evidence of parasite evasion in Stringmol. For our purposes: a ‘parasite’ is a molecule that is replicated, but is unable to replicate other molecules in return. ‘Parasite evasion’ is when *the container survives* the introduction of a

```

$BLUBO^B>C$=?>$ $BLUBO% }OYHOB
|
$BLUBO^B>C$=?G$ $BLUBO% }OYHOB

```

Figure 5: The functional regions of the replicase **R** (upper) and mutant **M** (lower). The location of the mutation is indicated by |. Both strings begin with the sequence ‘OBEQBXUTUDYGRHBBOREOLHHHRLUEUOBLROORE’ which is where the binding regions are located. The mutation has the effect of breaking the copy loop.

		Passive			
		<b>R</b>	<b>M</b>	<b>O</b>	<b>S</b>
Active	<b>R</b>	<b>R</b>	<b>M</b>	-	<b>S</b>
	<b>M</b>	<b>O</b>	<b>O</b>	-	<b>O</b>
	<b>O</b>	-	-	-	-
	<b>S</b>	-	-	-	<b>S</b>

Table 2: Interactions of: the replicase **R**; parasitic mutant **M**; product of the mutant **O**; the new strain of replicase that is immune to the parasite **S**. The body of the table shows the outcome of the reaction for each possible combination of active and passive molecules. Where the symbol ‘-’ appears instead of defined molecular species, it denotes no product formed.

parasite. We consider the container to have evaded the parasite when no parasitic molecules (of that strain) remain in the container. Parasite evasion was not originally detected and explained in [10], which is why we now provide an overview of the phenomena. We outline the two mechanisms by which Stringmol containers can survive a parasite and demonstrate that these mechanisms are not available in sticky-Stringmol.

We re-examined previously published Stringmol results [10] and located examples of mid-run parasites that were non-fatal to the container. Here, we give details of one such parasitic molecule and how it interacts with the dominant replicase. We use this example as the basis of our parasite evasion scenario. Figure 5 shows the functional region of the original replicase **R** and the parasitic mutant **M**. The parasite does not implement the loop in the microprogram that allows characters on the bound molecule to be iteratively copied. When the parasite **M** is the executing microprogram, the product of the reaction is **O**, a string of length one: ‘O’.

We examined how the Stringmol container survives this parasite in the original trial [10]. We found a new strain of replicase arose via a mutation in the binding region of **R**. This new strain **S** is never the executing molecule in reactions with **R** or **M** and is thus immune to the parasite, see table 2. The new strain that averts the death of the container, **S**, would have taken over the container via a ‘sweep’, even in the absence of a parasitic mutant as it is always passive when reacting with **R**. In the original trial, both **R** and **M**

Trial condition	No. Escapes
Stringmol	32
Stringmol no mutation	21
Sticky-Stringmol	0
Sticky-Stringmol no mutation:	0

Table 3: Number of escapes from the parasite scenario out of a possible 100 for the four trial conditions.

die out relatively quickly and the new strain becomes dominant. Figure 7 shows results of this scenario depicting typical dynamics of cases where this parasite is fatal and where the system evolves a resistant strain of replicase. Hence the container can sometimes evade what is a potentially fatal parasite.

We investigate the potency of the parasite in Stringmol in order to demonstrate that the parasite is potentially fatal. We also repeat this investigation for sticky-Stringmol and note it is unable to evade the parasite.

Our experimental setup initiates with two string types in the container: A replicase **R** and a parasitic mutant **M** of which there are 300 and 10 respectively at the start of each trial. The container is simulated until no molecules remain or until 0.5 million time steps. In each case we record if the container survived the parasite. We ran 100 trials for each of the four experimental conditions: Stringmol with and without mutation; sticky-Stringmol with and without mutation. The results are presented in table 3.

As we can see from table 3, it is possible for the Stringmol container to evade the parasite without mutation. In the absence of mutation it appears that the probability of binding between **R** and **M** being 0.66 is sufficient for the parasite to not to establish itself in the container in some cases, see figure 8 for typical dynamics. Stringmol (with mutation) is more successful at evading the parasite, see figure 7 for typical dynamics. There are two mechanisms by which the stringmol container can evade a (potentially fatal) parasite. One is by having a relatively low probability of binding, making it hard for new strains to establish themselves in the container. The second mechanism is the potential to mutate to a resistant strain of replicase. Looking at the results in table 3, it would appear that the dominant factor is the 0.66 chance of binding that the replicase, **R**, has with the parasite **M**.

Sticky-Stringmol appears unable to escape this parasite scenario with or without mutation. The ubiquitous binding at probability 1 causes the parasite to dominate the container every time. See figure 6 for dynamics that are typical of sticky-Stringmol both with and without mutation. Mutation in sticky-Stringmol offers no refuge from the parasite molecule. Because the binding is ubiquitous, a parasite is a parasite to all replicase molecules, rather than a limited subset of replicase molecules.



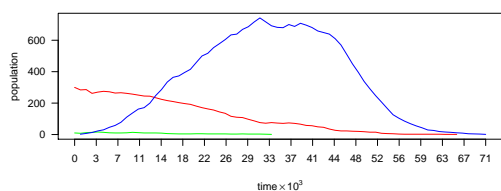


Figure 6: Sticky-Stringmol in the parasite evasion trial. The replicase, **R**, starts at the top at  $t=0$ . The mutant, **M**, starts at  $t=0$  and maintains a low population. The product of the mutation, **O**, peaks in the middle of the run. This figure is representative behaviour of all 200 runs (both with and without mutation).

These results have a bearing on the main point of the paper, the importance on non-trivial binding, which is that it is not only which molecules bind to which that is important. The probability of binding also plays a role in determining the system level properties. Reducing the probability of binding from 1 to 0.66 does not simply cause the same outcome to happen more slowly. This is a refinement on our previous comments and highlights a limitation of our approach to characterizing degeneracy, which requires a boolean understanding of molecular interactions.

## Discussion

Comparing the complexity of the artifacts in sticky-Stringmol with those of Stringmol, we note a loss of many of the more complex artifacts and no additional artifacts. These results demonstrate the importance of binding in AChems. They also indicate the potential for the complexity of a system to be stifled by a single naive component. This leads us to consider what other components of Stringmol (or any AChem) can have their levels of degeneracy measured and increased.

Investigations into the network properties of biological mutation networks, with an eye to how understanding their properties may lead to advances in ALife, are already underway [6]. That study makes use of network analysis techniques; our measure of degeneracy could be added to the array of such techniques. In cases where the sets  $A$  and  $B$  are the same, a binary interaction matrix specifies a network. Network analysis has a concept of ‘structural equivalence’ [15], which is the same as redundancy. The methods of measuring degeneracy and redundancy we introduce are also suitable for systems where  $A \neq B$ , which means they would also be applicable in other fields which do not have a natural mapping to a network, such as the binding of paratopes to epitopes in the immune system [2].

Much of the confusion regarding redundancy and degeneracy stems from the absence of a clear statement of context. A standalone statement of redundancy should take the form: ‘ $a_1$  and  $a_2$  are redundant given  $f$ , and in the context

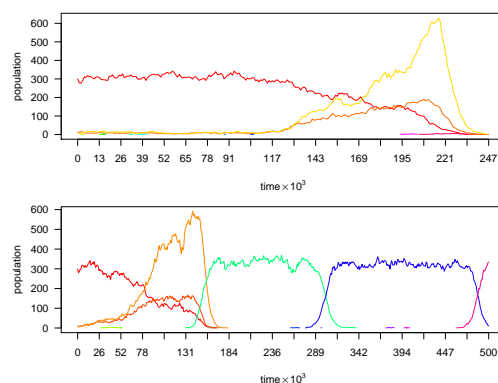


Figure 7: Stringmol (with mutation) in the parasite evasion trial. In both the upper and lower graphs the seed replicase, **R**, starts at the top at  $t=0$ . The upper graph shows a typical example of the dynamics when the parasite is lethal to the container. The 600 high spike towards the end of the run is the product of the parasite, **O**. The parasite, **M**, peaks at the same time as **O**, but to a height of only 200. The lower graph is an example where mutation gives rise to a new strain of replicase that is immune to the parasite and takes over the container. The 600 high spike is the product of the parasite, **O**. The parasite is fatal to the seed replicase **R**; but at the same time as the parasite and **O** are spiking, a new replicase molecule emerges. Typical Stringmol behavior can be seen for the remainder of the run, with two ‘sweeps’(where the dominant replicase is replaced by a mutation) occurring.

of  $B$ ’. Statements of the truncated form: ‘ $a_1$  and  $a_2$  are redundant’ are ambiguous, relying on the author and reader to have an identical understanding of both  $f$  and  $B$ . Under some alternative criteria,  $f'$  and/or  $B'$ , the elements  $a_1$  and  $a_2$  may well be redundant, degenerate or independent ( $B_{a_1} \cap B_{a_2} = \emptyset$ ). If an author states both  $f$  and  $B$  explicitly, then the context of the redundancy is captured unambiguously. When presented with an ambiguous statement, the best one can do is assume the statement is true and attempt to determine in what context(s) this is the case, as this may give valuable insight.

## Conclusion

We have introduced definitions of degeneracy and redundancy that can be applied to individual examples, such as ‘are two elements degenerate or redundant’, in equations 3 and 4. We have also introduced definitions of degeneracy and redundancy that can be applied when talking about the levels of these properties within a set, in equations 6 and 7. Our measures of degeneracy and redundancy of sets have been defined in such a way that the concept of degeneracy is not conflated with redundancy.

We applied these measures to the binding system used in Stringmol [10][11], demonstrating that the binding sys-



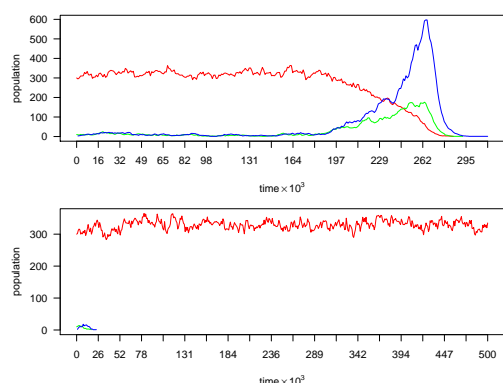


Figure 8: Stringmol without mutation in the parasite evasion trial. In both the upper and lower graphs the seed replicase, **R**, starts at the top at  $t=0$ . The upper graph shows a typical example of the dynamics when the parasite is lethal to the container. The 600 high spike towards the end of the run is the product of the parasite, **O**. The parasite, **M**, peaks at the same time as **O**, but to a height of only 200. The lower graph is an example of the parasitic mutant decaying out of the system, leaving the seed replicase, **R**, unaffected. The parasite **M**, and its product **O**, are present at  $t=0$ , but die out relatively quickly.

tem can produce both degeneracy and redundancy. We hypothesised the importance of a rich binding system to the complexity of the simulation artifacts Stringmol produces. We tested our hypothesis by constructing a deliberately poor Stringmol variant with ubiquitous binding, which we denote as ‘sticky-Stringmol’. Our measures show ubiquitous binding has no degeneracy. Our results demonstrated that a rich binding system facilitates many of the artifacts observed in Stringmol, including: hypercycles (two mutually dependent molecules), macro-mutations (non-point based mutations), sweeps (change of dominant replicase, other than by a random walk) [10], as well as parasite evasion. All of these complex phenomena were lost when we substituted Stringmol’s rich (degenerate) binding system for ubiquitous binding (no degeneracy).

We identified examples of parasite evasion in the previously published results of Stringmol [10], which we used as the basis of a parasite evasion experiment. We compared Stringmol to sticky-Stringmol in this parasite evasion trial, giving an overview of the mechanisms behind the loss of parasite evasion and demonstrating that the difference in binding system was the cause.

Though our hypothesis that ‘degeneracy in the components of an AChem facilitates the complexity of the system as a whole’ has not been proven in general terms, our results support this hypothesis and demonstrate the importance of binding systems in AChems.

## Acknowledgements

This work is part of the Plazmid project, funded by EPSRC grant EP/F031033/1

## References

- [1] W. Banzhaf, P. Dittrich, and H. Rauhe. Emergent computation by catalytic reactions. *Nanotechnology*, 7:307–314, 1996.
- [2] Melvin Cohn. An in depth analysis of the concept of polyspecificity assumed to characterize tcr/bcr recognition. *Immunologic Research*, 40:128–147, 2008.
- [3] James Decraene, George G. Mitchell, and Barry McMullin. Unexpected evolutionary dynamics in a string based artificial chemistry. In *ALife XI*, pages 158–165. MIT Press, 2008.
- [4] P. Dittrich, J. Ziegler, and W. Banzhaf. Artificial chemistries—a review. *Artificial Life*, 7(3):225–275, 2001.
- [5] Peter Dittrich and Pietro di Fenizio. Chemical Organisation Theory. *Bulletin of Mathematical Biology*, 69(4):1199–1231, 2007.
- [6] A. P. Droop and S. J. Hickinbotham. Properties of biological mutation networks and their implications for alife. *Artificial Life*, 2011 (in press).
- [7] Gerald M. Edelman and Joseph A. Gally. Degeneracy and complexity in biological systems. *PNAS*, 98(24):13763–13768, 2001.
- [8] Adam Faulconbridge, Susan Stepney, Julian F. Miller, and Leo Caves. RBN-world: The hunt for a rich AChem. In *ALife XII*, pages 261–268. MIT Press, 2010.
- [9] W. Fontana. Algorithmic Chemistry. In *ALife II*, pages 159–210. Addison Wesley, 1992.
- [10] Simon Hickinbotham, Edward Clark, Susan Stepney, Tim Clarke, Adam Nellis, Mungo Pay, and Peter Young. Diversity from a monoculture: Effects of mutation-on-copy in a string-based artificial chemistry. In *ALife XII*, pages 24–31. MIT Press, 2010.
- [11] Simon Hickinbotham, Edward Clark, Susan Stepney, Tim Clarke, Adam Nellis, Mungo Pay, and Peter Young. Specification of the stringmol chemical programming language version 0.1. Technical Report YCS-2010-457, Dept Computer Science, University of York, 2010.
- [12] Raphael Plasson, Axel Brandenburg, Ludovic Jullien, and Hugues Bersini. Autocatalyses. In *ALife XII*, pages 4–11. MIT Press, 2010.
- [13] G Tononi, O Sporns, and G M Edelman. A measure for brain complexity: relating functional segregation and integration in the nervous system. *PNAS*, 91(11):5033–5037, 1994.
- [14] Giulio Tononi, Olaf Sporns, and Gerald M. Edelman. Measures of degeneracy and redundancy in biological networks. *PNAS*, 96(6):3257–3262, 1999.
- [15] S. Wasserman and K. Faust. *Social Network Analysis: methods and applications*. Cambridge University Press, 1994.
- [16] James Whitacre. Degeneracy: a link between evolvability, robustness and complexity in biological systems. *Theoretical Biology and Medical Modelling*, 7(1):6, 2010.

# Evolving Three-Dimensional Objects with a Generative Encoding Inspired by Developmental Biology

Jeff Clune<sup>1</sup> and Hod Lipson<sup>1</sup>

<sup>1</sup>Department of Mechanical and Aerospace Engineering, Cornell University  
jeffclune@cornell.edu

## Abstract

This paper introduces an algorithm for evolving 3D objects with a generative encoding that abstracts how biological morphologies are produced. Evolving interesting 3D objects is useful in many disciplines, including artistic design (e.g. sculpture), engineering (e.g. robotics, architecture, or product design), and biology (e.g. for investigating morphological evolution). A critical element in evolving 3D objects is the representation, which strongly influences the types of objects produced. In 2007 a representation was introduced called Compositional Pattern Producing Networks (CPPN), which abstracts how natural phenotypes are generated. To date, however, the ability of CPPNs to create 3D objects has barely been explored. Here we present a new way to create 3D objects with CPPNs. Experiments with both interactive and target-based evolution demonstrate that CPPNs show potential in generating interesting, complex, 3D objects. We further show that changing the information provided to CPPNs and the functions allowed in their genomes biases the types of objects produced. Finally, we validate that the objects transfer well from simulation to the real-world by printing them with a 3D printer. Overall, this paper shows that evolving objects with encodings based on concepts from biological development can be a powerful way to evolve complex, interesting objects, which should be of use in fields as diverse as art, engineering, and biology.

## Motivation and Previous Work

The diversity, complexity, and function of natural morphologies is awe-inspiring. Evolution has created bodies that can fly, run, and swim with amazing agility. It would be desirable to harness the power of evolution to create synthetic physical designs and morphologies. Doing so would benefit a variety of fields. For example, artists, architects and engineers could evolve sculptures, buildings, product designs, and sophisticated robots. Evolution should be especially helpful in the design of complex objects with many interacting parts made of non-linear materials. In such challenging problem domains, evolution excels while human intuition is limited. Being able to evolve sophisticated morphologies also furthers biological research because it enables the investigation of how and why certain natural designs were produced. Evolving 3D objects is thus worthwhile both as a



Figure 1: Examples of evolved objects that were transferred to reality via a 3D printer.

basic science and for its innumerable potential applications. This paper describes how 3D shapes can be evolved and then transferred to reality via 3D printing technology (Figure 1).

Previous research in digital morphological evolution has typically involved encodings that were either highly biologically detailed, or highly-abstract with less biological accuracy. The former camp frequently simulates the low-level processes that govern biological development, such as the diffusing *morphogen* chemicals and proteins that determine the identity of embryonic cells (Bongard and Pfeifer 2001, Eggenberger 1997, Miller 2004). While this approach facilitates studying the mechanisms of developmental biology, the computational cost of simulating chemistry in such detail greatly limits the complexity of the evolved phenotypes. The most complex forms typically evolved in such systems are simple geometric patterns (such as three bands) (Miller

2004) or groups of shapes resembling the earliest stages of animal development (Eggenberger 1997).

The second camp employs high-level abstractions that enable the evolution of more elaborate forms with many parts, but these abstractions tend not to reflect the way that organisms actually develop (Wolpert and Tickle 2010, Bentley 1996). An example is Lindenmayer Systems (L-Systems), which iteratively replace symbols in strings with other symbols until a termination criteria is reached (Lindenmayer 1968, Hornby et al. 2003). While L-Systems can reproduce a wide variety of organismal shapes, especially those of branching plants, they do not model plant developmental processes (Wolpert and Tickle 2010). Another example is the work of Sims (1994), who evolved morphologies that resembled some biological creatures, although with an abstract encoding based on parameterized recursion that does not resemble natural developmental processes (Sims 1994).

A third option is possible, wherein a high-level abstraction is based on the developmental processes that give rise to natural forms. An example of this approach is Compositional Pattern Producing Networks (CPPNs) (Stanley 2007), which are used to evolve 3D objects in this paper and are described in Methods. Two groups have previously evolved 3D objects with CPPNs, although neither conducted an open-ended exploration of 3D objects. One group evolved CPPN objects that were composed of variable-sized spheres and were evaluated on two tasks: falling (Auerbach and Bongard 2010b) or moving rapidly (Auerbach and Bongard 2010a). Most of the evolved forms resembled clubs. A second group evolved soft-bodied robots to move quickly (Hiller and Lipson 2010). These studies demonstrate that CPPNs can create functional shapes, but leave open the question of what types of 3D objects CPPNs can produce with fewer constraints and without specific objectives.

2D pictures are evolved with CPPNs on *picbreeder.org*, where humans perform selection (Secretan et al. 2011). The complexity and natural appearance of the resulting images often support claims regarding the legitimacy of CPPNs as an abstraction of biological development (Stanley 2007). A demonstration in 3D would significantly strengthen these claims, however, because the natural world is 3D. It is possible that CPPNs are unable to frequently make sensible forms with the added difficulty of another dimension, and when objects must be one contiguous unit (which aids in transfers to reality). A recent paper by Bánsági Jr et al. (*Science* 2011) highlights the need to verify that generative encodings that produce complex patterns in 2D also can do so in 3D. By evolving CPPN objects in the natural 3D setting, this paper conducts a critical test of the hypothesis that generative encodings based on *geometric* abstractions of development capture some of the complexity-generating power of natural morphological development. Doing so also provides a visually intuitive testbed for studying how variants of such generative encodings behave. It also reveals the utility of

CPPNs as a representation for 3D object design.

## Methods

### Compositional Pattern Producing Networks

Compositional Pattern Producing Networks (CPPNs) abstract the process of natural development without simulating the low-level chemical dynamics involved in developmental biology (Stanley 2007). Cells (and higher-level modules) in natural organisms often differentiate into their possible types (e.g. heart or spleen) as a function of where they are situated in geometric space (Wolpert and Tickle 2010).

Components of natural organisms cannot directly determine their geometric location, so developmental processes have evolved to create gradients of chemicals and proteins called morphogens that organismal components use to figure out *where* they are and, thus, *what* to become (Wolpert and Tickle 2010). For example, in many animals the anterior-posterior and dorsal-ventral axes are specified by maternally provided morphogen gradients. Embryonic genes then construct more complicated geometric patterns of morphogens as a function of these simpler gradients. Downstream genes can construct additional pattern as a function of any of the patterns already created, enabling the production of patterns of arbitrary complexity (Wolpert and Tickle 2010).

CPPNs abstract this process by allowing similar geometric patterns to be composed of other geometric patterns, but represent the patterns mathematically instead of via diffusing morphogens. To replace maternally-provided gradients, the experimenter provides the initial gradients. Final patterns output by the CPPN determine the attributes of the phenotypic components at different geometric locations. For example, two-dimensional pictures could be encoded by iteratively passing the coordinates of each pixel on a canvas (e.g.  $x = 2, y = 4$ ) to a CPPN genome and having the output specify the color or shade of each pixel (Figure 2).

Each CPPN is a directed graph in which every node is itself a single function, such as sine or Gaussian. The nature of the functions can create a wide variety of desirable properties, such as symmetry (e.g. a Gaussian function) and repetition (e.g. a sine function) that evolution can exploit. Because the genome allows functions to be made of other functions, coordinate frames can be combined. For instance, a sine function early in the network can create a repeating theme that, when passed into the symmetrical Gaussian function, creates a repeating series of symmetrical motifs (Figure 2). This process abstracts the natural developmental processes described above (Wolpert and Tickle 2010).

The links that connect and allow information to flow between nodes in a CPPN have a weight that can magnify or diminish the values that pass along them. Mutations that change these weights may, for example, give a stronger influence to a symmetry-generating part of a network while diminishing the contribution from another part.

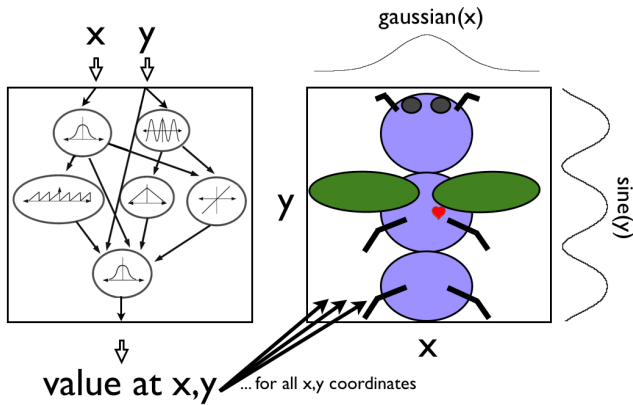


Figure 2: CPPNs combine mathematical functions to create regularities, such as symmetries and repeated modules, with and without variation. Adapted from Stanley (2007).

Variation is produced by mutating or crossing CPPNs. Mutations can add a node or change weights. The default set of allowable functions for CPPNs in this paper are sine, sigmoid, Gaussian, and linear, although we also experimented with additional functions (see Results). The evolution of the population of CPPN networks occurs according to the principles of the NeuroEvolution of Augmenting Topologies (NEAT) algorithm (Stanley and Miikkulainen 2002).

The NEAT algorithm contains three major components (Stanley and Miikkulainen 2002). (1) It starts with small genomes that encode simple networks and complexifies them via mutations that add nodes and links to the network. This complexification enables the algorithm to evolve the network topology in addition to its weights. (2) NEAT preserves diversity via a fitness-sharing mechanism that allows new innovations time to be tuned by evolution before competing them against more optimized rivals. (3) crossover utilizes historical information in a way that is effective, yet avoids the need for expensive topological analysis.

### Encoding 3D Objects with CPPNs

To evolve 3D objects, inputs for the  $x$ ,  $y$ , and  $z$  dimensions are provided to a CPPN. Additional gradients can be provided, which may bias the types of objects produced (see Results). A *workspace* (maximum object size) is defined with a *resolution*, which determines the number of voxels in each dimension. In this paper there are 10 voxels in the  $x$  and  $z$  dimensions and 20 in the  $y$  (vertical) dimension. The  $x$ ,  $y$ , and  $z$  value of each voxel are iteratively input to a CPPN, and voxels are considered full if the CPPN output is greater than a threshold (here set to 0.1), otherwise the voxel is considered empty. The 3D voxel array is then processed by the surface-smoothing Marching Cubes algorithm (Lorenson and Cline 1987). A normal is provided for each vertex when visualizing the objects in OpenGL, a graphics technique that further smooths the surface. These two smoothing steps en-

able high-resolution CPPN objects to be visualized without prohibitive computational costs.

This algorithm for encoding 3D objects is a more straightforward extension of how CPPNs encode 2D pictures (Stanley 2007, Secretan et al. 2011) than another algorithm for evolving 3D objects with CPPNs, which included growth over time and limited shapes to collections of attached spheres of different sizes (Auerbach and Bongard 2010b;a).

### Selection Mechanisms (Fitness Assignment)

We evolve images with interactive evolution and target-based evolution. During interactive evolution the user (here, the first author) views  $N$  rotating objects (here, 15) and selects a champion, which receives a fitness of 1000. The user can also reward additional organisms that receive a fitness of 500. To avoid uninteresting objects, those that are not chosen, yet have voxel counts between 10% and 90% of the maximum number possible, are given a fitness of 100. The remaining objects are given a fitness of 1. For target evolution, the fitness is the percent of voxels that matched the target object. To magnify differences in fitness values, all fitness scores serve as an exponent to a large constant  $c = 2000$  to produce the final fitness value. The parameters are identical to a previous work (Clune et al. 2011), except mutations were allowed to be larger (MutationPower = 2.5).

## Results and Discussion

### Interactive Evolution

**Overall summary** We study interactive evolution because it allows an open-ended exploration of the design space of objects CPPNs can produce. Additionally, interactive evolution avoids the greedy nature of target-based evolution, potentially allowing it to access more interesting objects (Secretan et al. 2011, Lehman and Stanley 2008). A drawback of interactive evolution is that it is subjective, but science should not abandon such a useful tool simply because it is subjective. While user preferences bias the types of objects selected, the encoding has to be able to produce such objects in the first place in order for them to be selected. Different encodings will bias the types of patterns evolved (Clune et al. 2011), meaning that interactive evolution can inform us about the biases and expressive power of the encoding.

Figure 3 shows example objects from different generations during a run of interactive evolution. The geometric patterns become more complex over generations, which reflects the property of complexification built into NEAT (Stanley and Miikkulainen 2002).

Figure 4 displays a few of the interesting objects discovered in different runs, some of which had different inputs and parameters (described below). It is important to note that these objects were chosen from a small number of runs performed by one person, each of which was limited to tens or perhaps a few hundred generations. It is noteworthy that such recognizable 3D forms emerge in such a



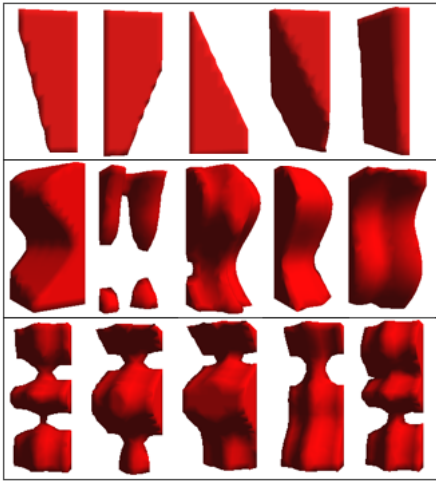


Figure 3: Representative objects from different generations of a single run of interactive evolution. From top to bottom, rows display individuals from generations 1, 15, and 33.

small sample size. These 3D objects should not be held to the same standard as pictures from *picbreeder.org*, where hundreds of users have published thousands of images after performing over 150,000 evaluations across hundreds of generations (Secretan et al. 2011).

The objects in Figure 4 exhibit many properties that are desirable both for studying morphological evolution and harnessing it for engineering or artistic purposes. The objects are frequently regular, a property which is important in engineering and for evolvability (Lipson 2007, Clune et al. 2011). An important regularity is symmetry, which is evident with respect to different dimensions in many of the objects. For example, all of the objects in generation 33 of Figure 3 are highly left-right symmetric, and objects b7 and b8 in Figure 4 exhibit left-right and top-bottom symmetries. Another useful regularity is repetition, which occurs frequently in the evolved objects (e.g. the top-right object in Figure 3). A further beneficial property is exhibiting regularity with variation (Stanley and Miikkulainen 2003, Lipson 2007, Clune et al. 2011). For example, Figure 4b1 has a motif that appears like an animal head, but is repeated in different sizes and with other subtle variations. Symmetric patterns with asymmetric variations can also be observed, such as in Figure 4a8 and Figure 4b6.

It is important to note that humans often select regular, symmetrical shapes, which increases their frequency in interactive evolution. That said, biology and engineering also often reward regularity. Additionally, it has been shown that when CPPNs generate artificial neural networks that control robots in target-based evolution, the neural wiring patterns are often regular, including symmetries and repeated themes (Clune et al. 2011), demonstrating that CPPNs produce regularities even without humans performing selection.

Most importantly, the evolved objects often look similar to natural forms or engineered designs, revealing that CPPNs can produce the types of objects we are interested in designing and studying with synthetic morphological evolution. Humans can only select such familiar forms if an encoding tends to produce such designs, which has not been the case for most previous generative encodings. People often describe Figure 4a2 and 4a3 as faces, 4a4 as a Jack-o'-lantern face, 4a5 as an animal figurine, 4a6 as an African statue of a human, 4a7 as a human female stomach, 4a8 as a human female torso, 4b1 and 4b4 as animals, 4b2 and 4b3 as elephants, 4b5 as a human head and shoulders, 4b6 as a horned mask, and 4b7 and 4b8 as spaceships. Some also describe 4b7 as a butterfly. People describe other objects as interesting art, even though they do not resemble any specific natural or human design (e.g. Figure 4a1). Such objects can potentially spark artistic ideas for new forms. The fact that the shapes consistently evoke human and natural designs demonstrates the expressive power of the CPPN encoding to produce interesting 3D objects.

An additional important property is that the offspring of the 3D CPPN objects are similar to their parents, but are varied in interesting ways. Some encodings lack this property in that mutations have dramatic effects, rendering most offspring very different from their parents, which hinders evolvability (Stanley and Miikkulainen 2003). For example, Figure 4b4 is the child of Figure 4b3, and Figure 4b2 is their close relative. All three are consistently described as animals, yet are interesting variations on the animal theme. For example, only a single generation of genetic changes between Figure 4b3 and Figure 4b4 transformed what appears like an elephant with a trunk into something resembling an elephant with warthog tusks. A different variant of Figure 4b3 that thickened the trunk can be seen in Figure 1 (center row, left), which is next to a printed copy of Figure 4b3. Moreover, Figure 4b3, its relative in Figure 1, and Figure 4b2 all evoke elephants, but they are quite different objects, suggesting that the CPPN has captured some fundamental aspects of the elephant concept that it expresses in different ways.

Some of the geometric complexity in the genome is not visible in these 3D phenotypes because a threshold determines the presence or absence of a voxel. In contrast, *picbreeder* pictures have a continuum of outputs in grayscale and color, which adds to their complexity. Pre-thresholded geometric information could be useful, however, to make colored 3D objects, or to have objects with multiple materials (e.g. the soft-robot equivalent of muscle and bone).

### Varying CPPN parameters generates different objects

To test whether the types of objects produced could be biased by the CPPN inputs and parameters, we performed multiple runs of interactive evolution with varying conditions. We initially provided only  $x$ ,  $y$ , and  $z$  values for

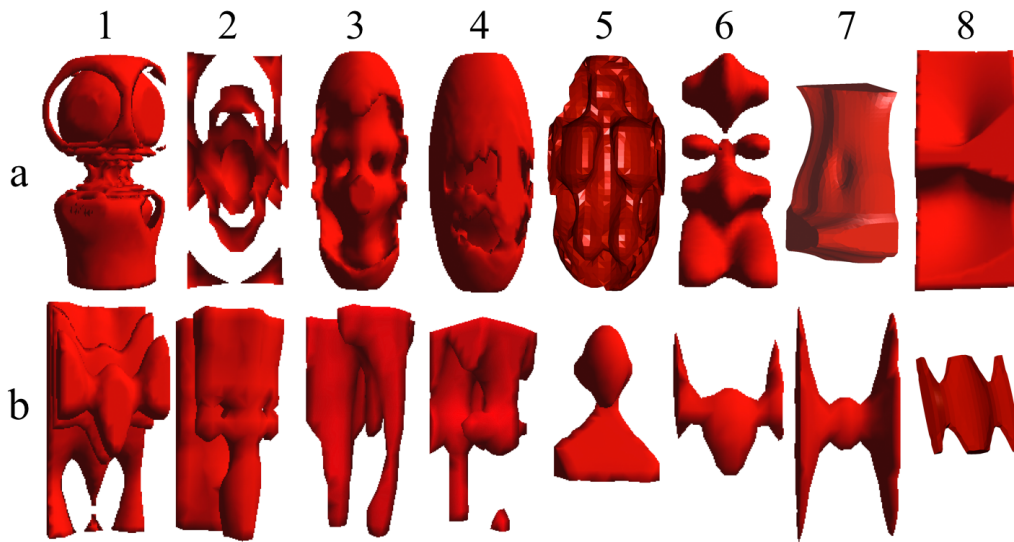


Figure 4: Example objects evolved with CPPNs via interactive evolution.

each voxel. Even with this minimal information, regularities such as symmetries and repeating themes were common (Figure 3), which is expected in a generative encoding with symmetric and repeating genomic functions. The objects in this setup seemed to require more generations before they became interesting, and usually did not appear like objects floating in space, but instead bordered the workspace wall.

We then added the distance from center as an input to the CPPN, which picbreeder also has (in 2D) (Secretan et al. 2011). This information more frequently created rounded objects centered in space. Because the distance-from-center function took the normalized values in each dimension, and the  $y$  (height) dimension was longer, an egg-shaped motif was common (Figure 5, left three). All of the objects in Figure 4 have this input. Preliminary experiments with other inputs also revealed interesting biases in the resulting objects (not shown), suggesting a rich area of research regarding how best to bias CPPNs with seed gradients.

To date, no published results explore how patterns differ when recurrence is allowed in CPPN genomes. We enabled recurrence and discovered that the resulting patterns are qualitatively different in that they tend to include fractal patterns. For example, branching patterns emerged, such as an object resembling a tree (Figure 6, left) and another evoking the vascular system (Figure 6, center). Like with fractals, the complexity is often concentrated at the surface boundary, producing a jagged surface effect (e.g. Figure 6, right). Objects with recurrent genomes were much more likely to have small, separated pieces floating in space.

Another interesting parameter of CPPNs is the set of possible genomic node functions. No research published to date has tested different function sets on the same problem to understand how CPPN patterns are affected by this pa-

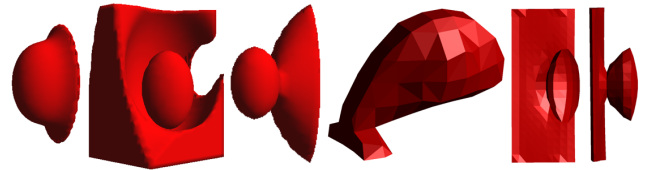


Figure 5: Objects evolved with a distance-from-center input (left three), which frequently featured egg-shape motifs, and objects evolved with an expanded set of genome functions (right three). The rightmost two images show different angles of the same object. Facets in the right three objects result from a close zoom and because, for illustration, normals are provided for facets instead of vertices.

rameter. Visual domains such as 3D objects are a helpful place to start such explorations because of the intuition they provide. We added a square, cosine, and sign-preserving square root function and performed additional runs. Objects in these runs tend to be more complex in earlier generations, and seem to involve both rounded and sharp edges. Figure 4b7 and the rightmost three in Figure 5 are example objects evolved with this expanded genomic node function set.

### Target-based Evolution

A second way to explore the capabilities of CPPNs is to challenge them to produce a target object. Knowing how CPPNs perform in 3D in target-based evolution is helpful for numerous reasons. Initially, it serves as a preliminary test of how CPPNs might perform on more open-ended, yet still target-based problems, such as evolving robot morphologies to perform certain tasks (e.g. locomotion). Additionally, biologists would benefit if they could repeatedly evolve var-

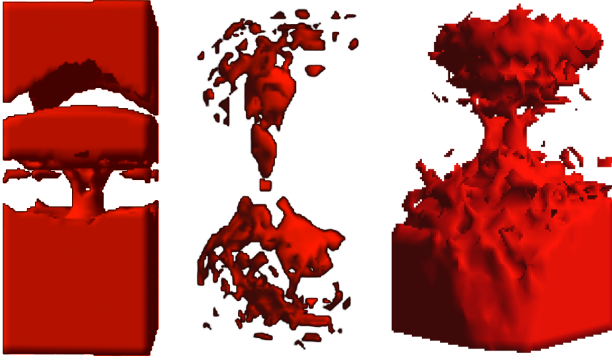


Figure 6: Example objects with recurrent genomes.

ious morphologies to study whether certain developmental strategies for constructing 3D geometric patterns arise frequently. Finally, target evolution allows an artist or engineer to explore objects that are similar to a target object, yet differ in interesting ways (similar to how Figure 4b4 and Figure 4b2 result from slight permutations to the genome of Figure 4b3). Finally, target-based evolution is much faster, enabling an exploration the effects of different parameter settings, which can inform interactive evolution.

The target object for this paper is shown in Figure 8a. It consists of four partially-overlapping spheres, with the outer two halved by workspace bounding box. This target has round shapes that are different from the egg-shaped motif facilitated by the distance-from-center input, providing a test of whether such a related input improves performance. Each treatment has 20 runs with a population of 150 for 1000 generations, unless otherwise specified.

The baseline treatment featured only  $x$ ,  $y$ , and  $z$  inputs and the default set of genome functions. The best performing object in each run captures the long cylindrical shape of the target, but most attempts at rounded edges are imperfect combinations of straight-line functions. All runs except one failed to carve much material away between the spheres. An average of 90.8% ( $\pm 0.003$  SE) of voxels are matched (Figure 7), but the target object is not identifiable until about  $\geq 93\%$  of voxels are matched. As such, the small differences in fitness between the treatments in Figure 7 represent substantial differences in whether the target object is recognizable. Interestingly, one outlier run in this treatment performed much better than the rest (with 94.6% of voxels correct). It features rectangular approximations of spheres (Figure 8b). The lack of round shapes in this treatment corroborates the previous subjective observation from interactive evolution that CPPNs can struggle to evolve and exploit round gradients when they are not provided as inputs.

To test if seeding CPPNs with spherical gradients makes it easier to match this rounded target, we added distance to the center as an input. The CPPNs in the previous treatment

could have evolved to calculate this same information, but that may have been difficult. Surprisingly, this information significantly lowered performance to 90.0% ( $\pm 0.002$  SE,  $p = 0.013$ , Mann-Whitney test, Figure 7). However, the evolved objects all have smooth, round forms (Figure 8c-d), confirming that providing different seed gradients can bias the types of evolved objects. While this might be expected in early generations, it is interesting that the gradients provided have noticeable effects after a thousand generations. This result is in line with a previous paper that found that the information input into CPPNs can bias the resulting phenotypes (Clune et al. 2009). We include this input in the remaining treatments in this paper because it facilitates round surfaces, even though it hurt performance in this experiment.

Because interactive evolution features smaller population sizes, it is worthwhile to study how this difference affects the search for 3D objects. Additionally, since NEAT complexifies genomes over evolutionary time, having more generations may improve the search by accessing genomes with more hidden nodes. We investigate these issues by decreasing the population size from 150 to 15 and increasing the number of generations tenfold to  $10^4$ , which keeps the number of evaluated objects the same. This change significantly improves performance to 91.8% ( $\pm 0.003$  SE,  $p < 0.001$ , Mann-Whitney test, Figure 7), suggesting that the small population sizes in interactive evolution do not hurt, and may actually benefit, morphological evolution with NEAT-based encodings. The evolved objects tend to have more space carved out between the spheres (Figure 8e-f).

A fundamental evolutionary parameter that can greatly affect evolvability is the mutation rate. We varied the major sources of mutation in NEAT by altering the rate at which genomic links are added, removed, and mutated, as well as the rate at which genomic nodes are added. Increasing the node addition rate significantly boosted performance ( $p < 0.001$ , Mann-Whitney test, Figure 7) to 91.5% ( $\pm 0.003$  SE). Changing the other mutation rate parameters did

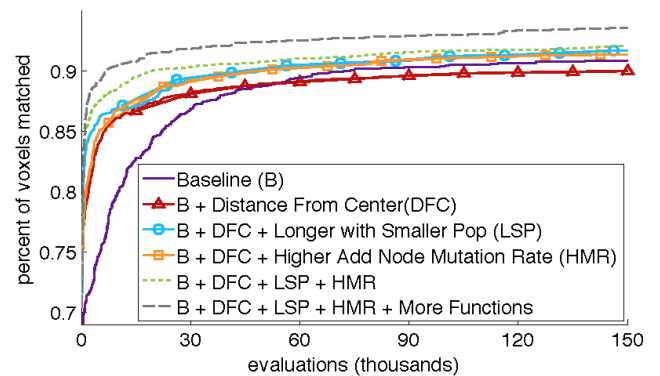


Figure 7: Means of the best-performing individuals for target-based evolution. See text for variance.

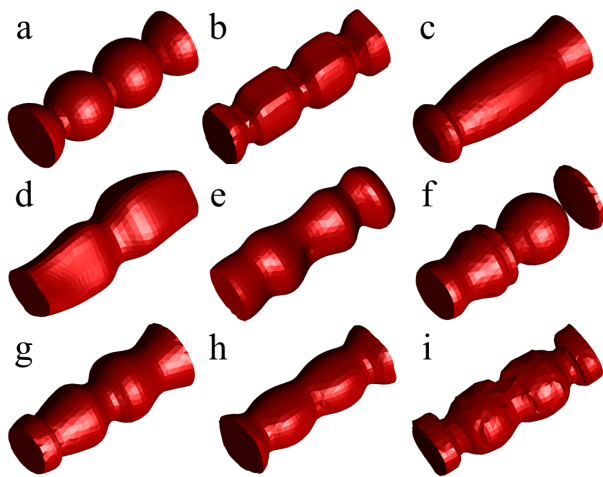


Figure 8: Target-based evolution objects.

not improve performance (data not shown).

Because a smaller population with more generations was beneficial, and because a higher mutation rate was beneficial, we tested whether both changes together would outperform either alone. The combination did improve performance to 92.0% (Figure 7), but the difference was not significant ( $p > 0.05$ , Mann-Whitney test). We also found that the expanded genome function set (described previously) improved performance to 93.0%, which was significant ( $p = 0.022$ , Mann-Whitney test). As before, the objects in this treatment seemed to combine rounded surfaces with sharper edges: while most were smooth (e.g. Figure 8g-h), a few had rough patches on their surface, including Figure 8i. Adding recurrent genomic connections to this treatment did not significantly affect performance (93.3%,  $p > 0.05$ ).

Overall, the target-based evolution experiments reveal that evolving CPPNs can roughly match a target object. While a high percentage of voxels were matched, the degree to which the evolved objects qualitatively resemble the target is subjective and debatable. The most important contribution of these experiments is to better understand the way in which target-based evolution is biased by different parameters. These results are preliminary, however, until more tests can be conducted with additional targets.

It is also interesting that many of the evolved objects look designed for a purpose. For example, many of the objects in Figure 8 seem like functional and aesthetically attractive objects carved on a lathe, such as legs from tables and chairs or posts from banisters and railings. One reason this is surprising is because it could have been the case that the greedy nature of target-based evolution would have gained improvements by iteratively adding small patches of voxels that match a subset of the overall space. Such a patchwork solution would not look as regular and smooth as the objects that actually evolved, suggesting that CPPNs are bi-

ased away from such a piecemeal strategy. Previous work has shown that CPPNs have difficulty making exceptions to regular patterns when evolving neural networks (Clune et al. 2011), which could explain why the target object in this study was not matched one patch at a time. Such a bias toward regularity may simultaneously explain the smoothness of the evolved objects and why matching the final few percent of voxels is so difficult.

Artists and engineers may actually benefit from the fact that the evolved objects share some properties of the target, but are different in interesting ways. This means that a designer can provide a seed object as a target, and a series of objects can automatically be generated that are aesthetically interesting variations on that seed concept (Figure 8).

## Transferring Objects to the Physical World

Advances in 3D printing technologies make it possible to transfer evolved objects into the physical world, which may help artists and engineers benefit from this technology. To test whether CPPN objects maintained their appearance and structural integrity in reality we printed them on a Connex500 3D printer. The objects look similar to their simulated counterparts and are structurally sound (Figure 1). One difference is that non-contiguous pieces (e.g. the top of Figure 6, left) are not held in place in the physical world without additional scaffolding. By printing in a semi-transparent material, we also discovered that none of the objects have visible hollow areas embedded within them, although CPPNs can create such negative spaces. While the gap between simulated and physical objects was not expected to be large for static objects, it is helpful to have verified the fidelity of the transfer.

## Conclusions and Future Work

This paper introduces an algorithm for evolving 3D objects with the CPPN generative encoding, which is a computationally efficient abstraction of biological development. We conducted both interactive and target-based evolution to explore the ability of CPPNs to create complex objects, especially those that resemble natural and engineered designs.

A small, preliminary exploration of the design space of 3D CPPN objects unearthed a diversity of objects that evoke natural and engineered forms. Many of the objects featured regularities such as symmetry and repetition, with and without variation. Such properties are important for engineering and evolvability (Lipson 2007, Clune et al. 2011), and suggest that CPPNs are a promising encoding for evolving useful and aesthetically pleasing objects. To extend this research we are creating a website like picbreeder.org (Secretan et al. 2011) where users can collaboratively evolve 3D objects online, which will provide a much larger exploration of the potential of this technology. It will also overcome the need for any individual to perform all of the evaluations in a lineage and thus allow more complex objects to evolve.



Experiments with target-based evolution on one target revealed how the inputs and parameters of CPPNs can influence the types of objects they evolve. The evolved objects roughly resemble the target, but do not match it precisely. While the evolved objects share some properties of the target, they also differ from it in interesting ways. This property could help artists and engineers by providing 3D designs that are variations on a seed concept. All of these conclusions are tentative, however, since experiments were only conducted with one target. Future work is necessary to determine whether these observations generalize.

While there are many useful applications for evolving static, single-material 3D objects, this technology is also a stepping stone to evolving objects that can move and that have multiple materials. In future work we will evolve such soft-bodied robots in simulation and transfer them to the physical world. Doing so will enable us to harness the power of evolution and developmental biology to begin to create synthetic creatures that have some of the exciting properties of their natural counterparts.

### Acknowledgments

Thanks to Jon Hiller, an NSF Postdoctoral Research Fellowship in Biology to JC (DBI-1003220), NSF CDI Grant ECCS 0941561, and NSF Creative-IT Grant IIS 0757478.

### References

- Auerbach, J. and Bongard, J. (2010a). Dynamic resolution in the co-evolution of morphology and control. In *Proceedings of Artificial Life XII*.
- Auerbach, J. and Bongard, J. (2010b). Evolving CPPNs to grow three-dimensional physical structures. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 627–634. ACM.
- Bentley, P. J. (1996). *Generic Evolutionary Design of Solid Objects using a Genetic Algorithm*. PhD thesis, University of Huddersfield.
- Bongard, J. and Pfeifer, R. (2001). Repeated structure and dissociation of genotypic and phenotypic complexity in artificial ontogeny. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 829–836.
- Clune, J., Ofria, C., and Pennock, R. (2009). The sensitivity of HyperNEAT to different geometric representations of a problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 675–682.
- Clune, J., Stanley, K., Pennock, R., and Ofria, C. (2011). On the performance of indirect encoding across the continuum of regularity. *IEEE Transactions on Evolutionary Computation*, To appear.
- Eggenberger, P. (1997). Evolving morphologies of simulated 3D organisms based on differential gene expression. In *Fourth European Conference on Artificial Life*, pages 205–213. The MIT Press.
- Hiller, J. and Lipson, H. (2010). Morphological evolution of freeform robots. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 151–152. ACM.
- Hornby, G., Lipson, H., and Pollack, J. (2003). Generative representations for the automated design of modular physical robots. *IEEE Transactions on Robotics and Automation*, 19(4):703–719.
- Lehman, J. and Stanley, K. (2008). Exploiting open-endedness to solve problems through the search for novelty. *Artificial Life*, 11:329.
- Lindenmayer, A. (1968). Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. *Journal of Theoretical Biology*, 18(3):280–299.
- Lipson, H. (2007). Principles of modularity, regularity, and hierarchy for scalable systems. *Journal of Biological Physics and Chemistry*, 7(4):125.
- Lorensen, W. and Cline, H. (1987). Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169.
- Miller, J. (2004). Evolving a self-repairing, self-regulating, French flag organism. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 129–139. Springer.
- Secretan, J., Beato, N., D’Ambrosio, D., Rodriguez, A., Campbell, A., Folsom-Kovarik, J., and Stanley, K. (2011). Picbreeder: A Case Study in Collaborative Evolutionary Exploration of Design Space. *Evolutionary Computation*, To appear.
- Sims, K. (1994). Evolving 3D morphology and behavior by competition. *Artificial Life*, 1(4):353–372.
- Stanley, K. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162.
- Stanley, K. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.
- Stanley, K. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130.
- Wolpert, L. and Tickle, C. (2010). *Principles of Development*. Oxford University Press, 4th edition.

# Selective pressures for accurate altruism targeting: evidence from digital evolution for difficult-to-test aspects of inclusive fitness theory\*

Jeff Clune<sup>1</sup>, Heather J. Goldsby<sup>1</sup>, Charles Ofria<sup>1</sup>, and Robert T. Pennock<sup>1</sup>

<sup>1</sup>Michigan State University  
jclune@msu.edu

## Abstract

Kin selection theory predicts that evolution favors altruist genes that are more accurate in targeting altruism only to copies of themselves. We support this prediction by competing multiple altruist-targeting mechanisms that vary in accuracy in determining if a recipient carries a copy of the altruist gene. We compete altruism-targeting mechanisms that make energy donations based on (1) kinship (kin targeting), (2) genetic similarity at a level greater than expected for kin (similarity targeting), and (3) perfect knowledge of the presence of an altruist gene (Green Beard targeting). Natural selection favored the most accurate targeting mechanism available, once altruism *levels* were accounted for (Fig. 1). Our investigations also revealed that the Green Beard mechanism, originally invented as a hypothetical example of a perfectly accurate, cheater-proof system and subsequently discovered in nature, is in fact vulnerable to cheaters. Such cheaters prevent Green Beard targeting from outcompeting kin and similarity targeting (Fig. 1c). The reason is because Green Beard altruists donate to organisms that have Green Beards and make *at least one* Green Beard donation. There is thus an evolutionary pressure to donate only once, thereby qualifying to receive Green Beard donations while paying as little as possible. By increasing the number of donations necessary to *qualify* to receive Green Beard donations ( $T$ ), we showed that organisms evolved to donate just above this threshold (Fig. 2). Green Beard targeting could only take advantage of its increased accuracy and outcompete kin and similarity targeting when we artificially set  $T$  to a high number, such as 100 (Fig. 1d). These results raise the question of how kin and similarity targeting differ from Green Beard targeting in being able to raise altruism levels despite the presence of cheaters. The answer is that they have built-in mechanisms that keep cheaters at bay (Fig. 3). We propose that Green Beard targeting can be augmented with a similar defense against cheaters if mutations that change the altruism level also change the marker (e.g., beard color), such that beard color reliably indicates the altruism level. This *Identical Beard Color* mechanism raises its altruism level automatically and outcompetes kin and similarity targeting due to better accuracy (Fig. 1e). Overall, our results confirm that natural selection favors altruist genes that are increasingly accurate in targeting altruism to only their copies. Our work also emphasizes that the concept of targeting accuracy must include both the presence of an altruist gene and the *level* of altruism it produces.

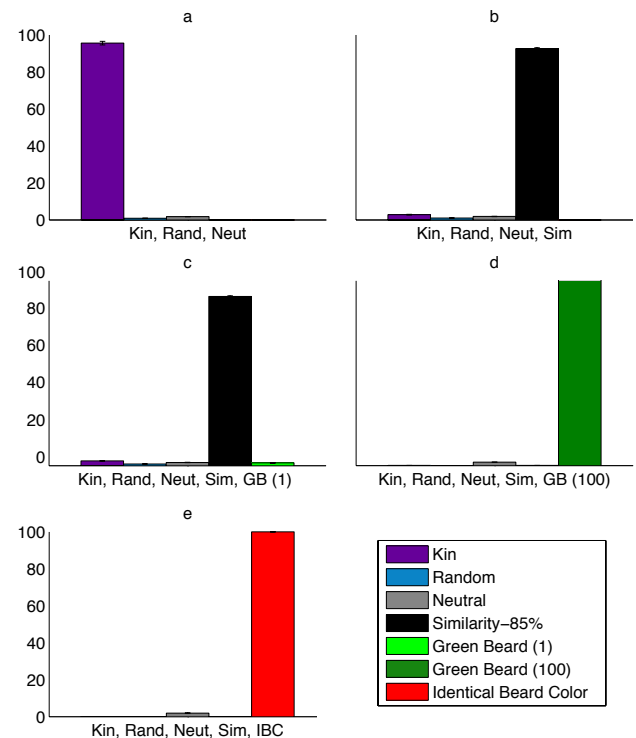


Figure 1: Evolved altruism levels for different targeting mechanisms. Plotted is the average number of donations made by last-generation organisms in 50 trials ( $\pm$  one standard error, often too small to distinguish). The maximum number of donations is capped at 100. (a) Targeting altruism based on kinship was selected for over two controls (targeting altruism at random, and a neutral instruction). (b) Targeting altruism based on high genetic similarity was favored over targeting based on kinship. (c) Selection did not favor targeting altruism via a Green Beard mechanism (with a threshold of 1) over kin and similarity targeting. (d) Selection favored a Green Beard mechanism with a threshold of 100 (the maximum number of donations allowed) over kin and similarity targeting. (e) Selection favors Identical Beard Color targeting over kin and similarity targeting.

\*Published in Proc. Royal Society, 2011, 278(1706): 666-674.

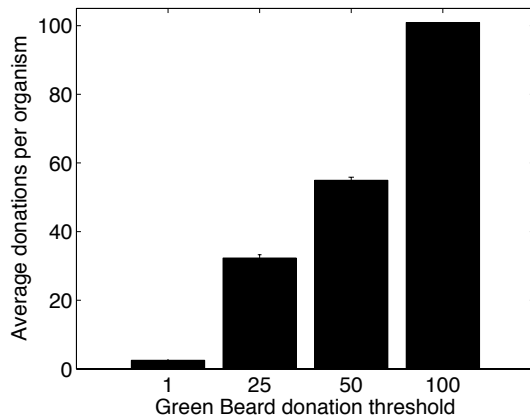


Figure 2: Evolved altruism levels for different Green Beard thresholds. Plotted is the average number of donations made per organism for different threshold values ( $T$ ) of the donate-threshold-gb instruction (averaged from the final populations of 50 trials per treatment  $\pm$  one standard error, often too small to distinguish). Organisms evolved to perform enough donations to surpass the threshold and thus qualify to receive altruism, but did not perform substantially more than  $T$  donations.

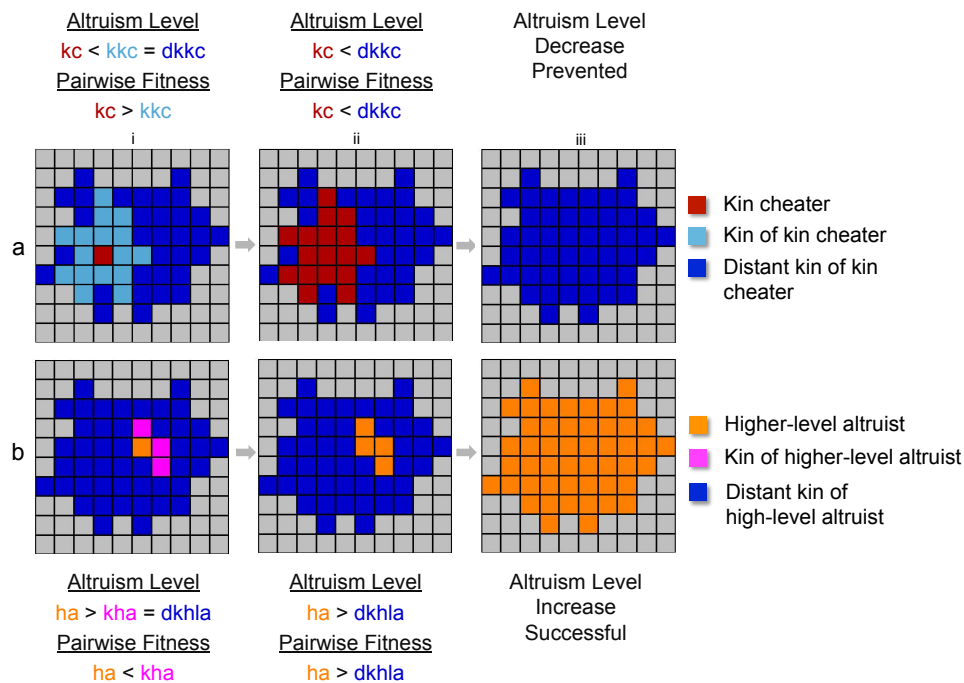


Figure 3: How kin and similarity targeting can evolve persistently high altruism levels. A thought experiment illustration showing how (a) kin-based altruism naturally thwarts kin-cheaters and (b) enables enduring increases in altruism levels. (a-i) Consider a group of related organisms that are altruistic to each other (blue and light-blue). One organism may mutate to be less altruistic, becoming a kin-cheater (red), but since only its closest relatives (light-blue) will consider it kin, only they will be altruistic toward it. (a-ii) The kin-cheater will tend to supplant its kin because it receives more donations from them than it gives. (a-iii) Once the kin-cheater has replaced those that considered it kin, the kin-cheater is left receiving donations only from other kin-cheaters. This group (red) will have a lower altruism level than their distant kin (blue) and will come to be replaced by them. (b-i) Now consider an organism (orange) that mutates to have a higher level of altruism than its ancestors (blue). Initially, it will be selected against because it gives more donations to those that it considers kin (pink) than it receives from them. (b-ii) If the less-altruistic kin of the higher-level altruist are killed off by drift, then the higher-level altruist and its offspring (orange) will have a competitive advantage over their distant ancestors (blue). (b-iii) While chance is required to start the process, once it has occurred there will be selection for the higher level of altruism. There are additional factors that complicate all of these fitness comparisons, but for clarity we have sketched these scenarios only in broad strokes.

# Natural Selection Fails to Optimize Mutation Rates for Long-Term Adaptation on Rugged Fitness Landscapes\*

Jeff Clune<sup>1,2</sup>, Dusan Misevic<sup>3</sup>, Charles Ofria<sup>1</sup>, Richard E. Lenski<sup>1</sup>, Santiago F. Elena<sup>2</sup>, and Rafael Sanjuán<sup>2</sup>

1. Michigan State University 2. Universidad Politecnica de Valencia. 3. ETH Zurich  
jclune@msu.edu

## Abstract

The rate of mutation is central to evolution. Mutations are required for adaptation, yet most mutations with phenotypic effects are deleterious. As a consequence, the mutation rate that maximizes adaptation will be some intermediate value. This abstract summarizes a previous publication in which we used Avida, a well-studied artificial life platform, to investigate the ability of natural selection to adjust and optimize mutation rates. Our initial experiments occurred in a previously studied environment with a complex fitness landscape (Lenski et al. *Nature*, 423, 2003) where Avidians were rewarded for performing any of nine logic tasks. We assessed the optimal mutation rate by empirically determining which unchanging mutation rate produced the highest rate of adaptation. Then, we allowed mutation rates to evolve and we evaluated their proximity to the optimum. Although we chose conditions favorable for mutation rate optimization (asexual organisms not yet adapted to a new environment), the evolved rates were invariably far below the optimum across a wide range of experimental parameter settings (Fig. 1). We hypothesized that the reason mutation rates evolved to be suboptimal was the ruggedness of fitness landscapes. To test this hypothesis, we created a simplified ‘counting ones’ (a.k.a. ‘onemax’) landscape without any fitness valleys and found that, in such conditions, populations evolved near-optimal mutation rates (Fig. 2, top row). In contrast, once moderate fitness valleys were added to this simple landscape, the ability of evolving populations to find the optimal mutation rate was lost (Fig. 2, bottom two rows). Additional experiments revealed that lowering the rate at which mutation rates evolved did not preclude the evolution of suboptimal mutation rates (see original manuscript). We conclude that rugged fitness landscapes can prevent the evolution of mutation rates that are optimal for long-term adaptation because of the short-term costs of traversing fitness valleys. This finding has important implications for evolutionary research in both biological and computational realms.

\*Published in PLoS Computational Biology, 2008, 4(9).

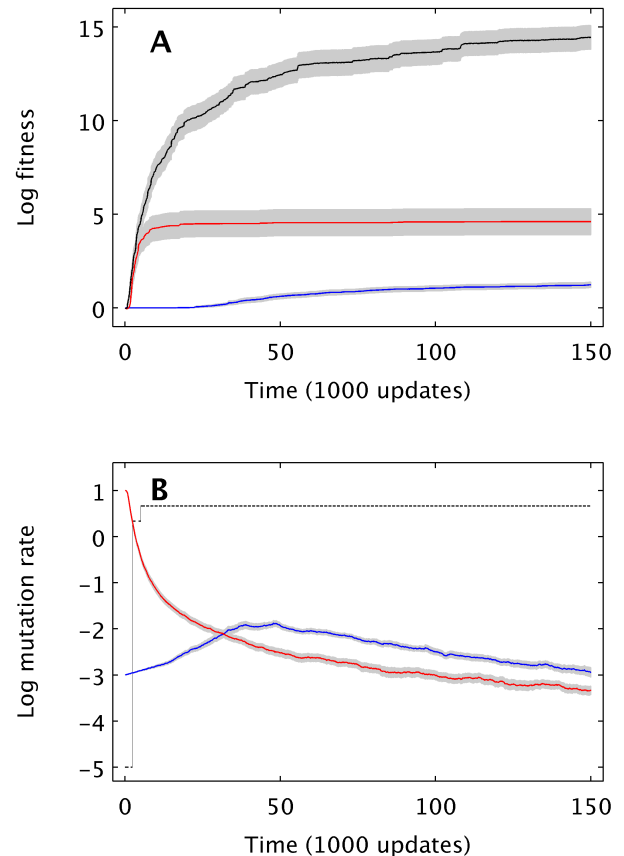


Figure 1: Evolutionary trajectories for fitness and mutation rate on a complex fitness landscape reveal that evolved mutation rates are lower and produce less adaptation (lower fitness values) than a certain (long-term optimal) non-evolving rate. (A) Evolution of average (over 50 runs) log-fitness  $\pm 1$  s.e.m. for treatments with the genomic mutation rate fixed at the empirically determined optimum rate  $U_{opt} = 4.641$  (black) and for treatments with variable, evolving genomic mutation rates starting at either 10 (red) or  $10^{-3}$  (blue). (B) Evolution of average log genomic mutation rate  $\pm 1$  s.e.m. for treatments with variable, evolving mutation rates starting at either 10 (red) or  $10^{-3}$  (blue). The black line indicates the mutation rate that had produced the highest average fitness for that time point.

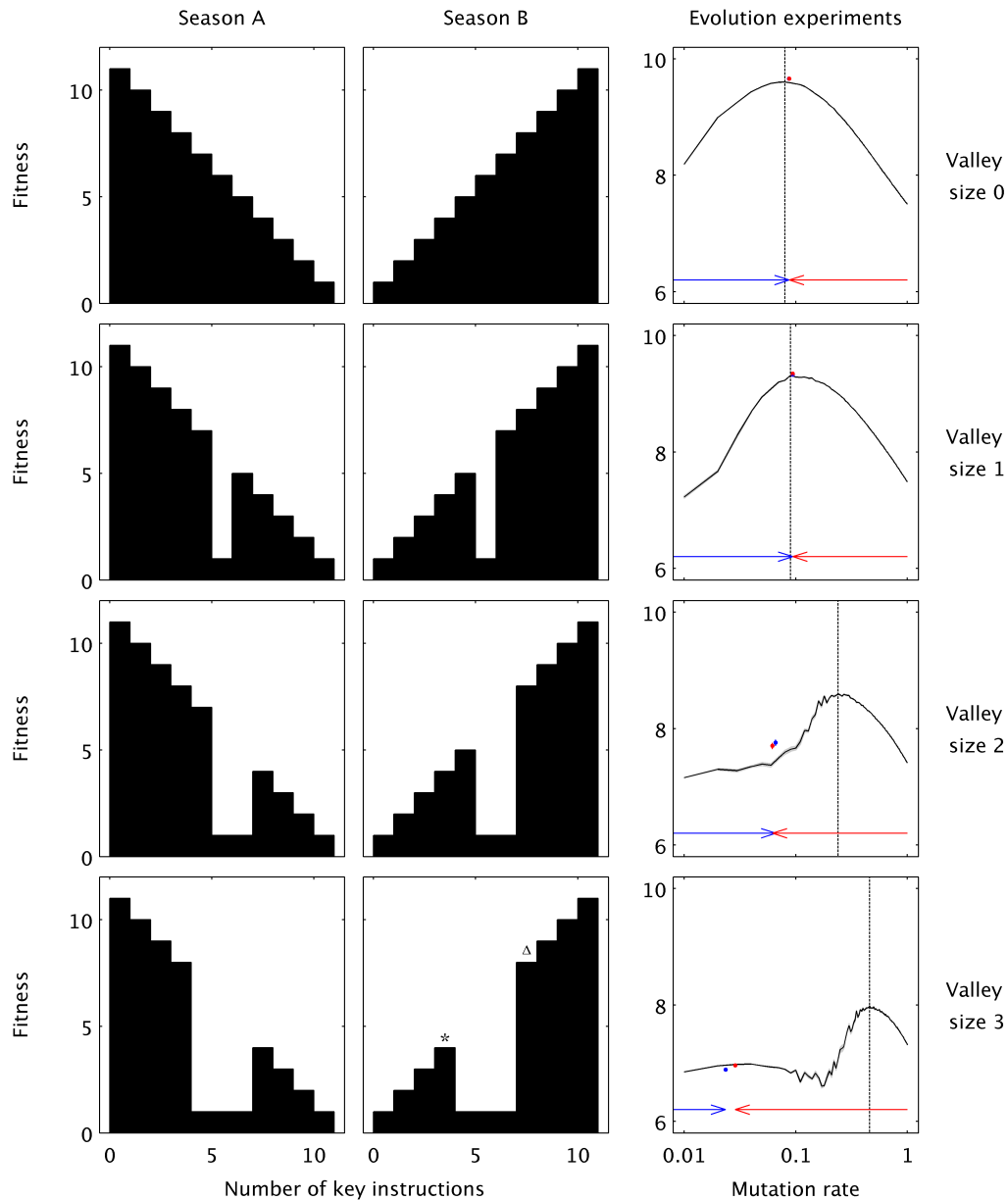


Figure 2: Evolution of mutation rates on simple fitness landscapes with different ruggedness. Here, fitness depended solely on the match between the environment and the number of a key instruction that organisms had in their genomes. In season A (left column) the key instruction was deleterious while it was beneficial in season B (center column). Rugged fitness landscapes with maladaptive valleys (rows 2-4) were introduced by setting the fitness of organisms with intermediate numbers of the key instruction to the minimum fitness level of one. The right-most column shows the results of evolution experiments under each of these selective regimes. Final fitness is shown as a function of genomic mutation rate for both static and dynamic mutation rates. The solid black line represents the average of the mean fitness across 10 runs for each of 100 different static mutation rates ranging from  $U = 0.01$  to 1 in increments of 0.01. The two colored points represent the mean fitness and mutation rate, both averaged over 50 runs where the mutation rate freely evolved, with initial rates of  $U = 1$  (red) or  $10^{-5}$  (blue). Mutation rate and fitness values were time-averaged over the last 10 of 50 environmental changes. Owing to very similar final values, despite the very large initial differences, the individual colored points are indistinguishable in the first two rows, and error bars are not visible. The arrows indicate where mutation rates began and ended, on average, for the dynamic-rate experiments. Although the optimal mutation rate increases as a function of valley size (note the right-shift in the dashed line from top to bottom), the evolved mutation rates in fact decrease as a function of valley size (note the left-shift of the blue and red points from top to bottom).

# Recovering Hidden Swarm Parameters Using a Simulated “Robofish”

Richard Coates<sup>1</sup> and Simon Hickinbotham<sup>1</sup>

<sup>1</sup>YCCSA, Department of Computer Science, University of York, Heslington, York YO10 5GH, UK  
rdc501@gmail.com      sjh@cs.york.ac.uk

## Abstract

Swarming is behaviour which emerges from the action of individual agents. Models of swarm behaviour impose fixed model parameters on the agents comprising the swarm. This paper evaluates the possibility of extracting the parameters of a swarm model from the swarm. This can be achieved by evolving the parameters of a single agent that interacts with the swarm. The approach was inspired by work on so-called “robofish” by Faria et al. If we assume that the collective dynamics of wild animals can be modelled, it would be desirable to recover the dynamics of the model via interaction with them. We demonstrate that it is possible to recover the parameters of a shoaling model used by a swarm. We present an evaluation of this approach, using a genetic algorithm to drive the learning process. The experiments also reveal information about the effects of varying the parameters of the model on the emergent swarm dynamics.

## Introduction

In nature many animals travel in flocks, shoals, swarms and other large groups. Several models have been proposed to replicate this phenomena (Aoki, 1982; Reynolds, 1987; Couzin et al., 2002). In each of these models, individuals follow local rules which produce the swarm as an emergent phenomenon. However little work has been done to find out the validity of these models and how accurately they map to behaviour in real-world swarms. In this paper we explore this possibility by investigating whether an agent can recover the parameters of a swarm by monitoring its interactions with the rest of the swarm. We test an evolutionary approach to this learning problem. If the evolved behaviour of the agent and swarm is identical then the models used by them are functionally equivalent. This novel approach was inspired by the work of Faria et al. (2010), in which a robotic fish (or *robofish*) interacts with a shoal of sticklebacks *Gasterosteus aculeatus* L. We refer to the shoal of fish with which the robofish interacts as the *modelfish* in the rest of this paper, since it is presumed that the shoal is following a model of behaviour which the researcher is trying to discover. This paper extends the work presented in (Coates and Hickinbotham, 2011).

It is only possible to determine whether model parameters can be learned in this way if the parameters of the swarm are known. Accordingly, we test the approach in simulation, where a swarm of modelfish follow a pre-specified model. In addition, the fitness function which is used to evolve the robofish model only makes use of observations about the emergent behaviour of the swarm. In other words, the model parameters are “hidden” from the evolutionary algorithm.

Aoki (1982) described one of the first attempts to accurately model the behaviour of fish. The behaviour of each fish is determined by the position of its neighbors. The space around the fish is divided into zones of perception relative to the position and orientation of the fish. Each zone has a corresponding behaviour linked to it, commonly called a *compulsion*. The presence of fish in a particular zone increases the contribution of the compulsion to the behaviour of the fish in the next discrete time step. In Aoki’s model, the fish’s area of perception is split into three radii. Behind the fish there is also a “blind spot”, the contents of which do not contribute to the fish’s behaviour. Nearest the fish is the zone of repulsion, then comes the zone of orientation and finally the zone of attraction. Aoki’s model only takes into account up to four neighboring fish in the zone of perception. These are selected randomly (with a greater chance of choosing those in front) and have a diminishing effect of the movement of the fish. If no neighbours are found in the zone of perception, the fish will try and move towards any it can see, no matter the distance. The neighbours it finds (and the zones they appear in) change the mean and standard variation of a Gaussian model which governs the update of the heading of the fish.

Swarm simulations were originally developed to study the behaviour of real animals. They were used to provide insights into possible reasons for why swarms behave as they do and how changing the simulations parameters affects the resulting behaviour. Since then they have found uses in computer graphics (Reynolds, 1987), as well as in searching data (Kennedy and Eberhart, 1995). Reynolds (1987) developed one of the best known swarm models, designed not necessarily to be authentic but to give aesthetically pleasing move-

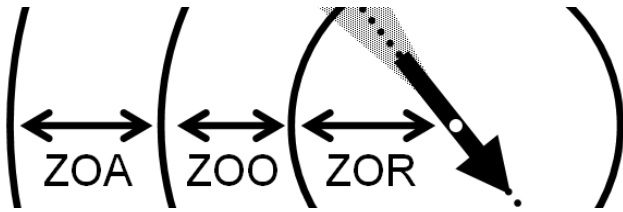


Figure 1: The three zones of the 2D Couzin model. The orientation of the fish is shown by a bold arrow (centre). The blind spot is shown as a shaded area. The widths of the zones of Attraction (ZOA), Orientation (ZOO) and Repulsion (ZOR) are indicated by arrows. Note that the position of the outer zones relative to the fish are dependent upon the widths of the inner zones.

ment for animation purposes. It consists of a single zone of perception within which a “boid” is able to detect others. Its movement is also governed by three compulsions with regards to the neighbours it can see, the urge to move towards a neighbour, away from them or in the same direction as them. These compulsions are combined with the position and heading of the neighbours to provide a new heading for the boid.

The Couzin model shares much with Aoki’s model. It retains the three zones and their effect on the heading of the fish. However like Reynolds’ model fish outside of the range of perception have no effect at all. In addition, in a single time step, only one of the zones ever has an influence on the behaviour of the fish. The zone of repulsion (ZOR) takes precedence over the other two compulsions. If a neighbour is found in this zone then the other two are ignored and the fish will just try and swim away from neighbours in the ZOR. If no fish are found in the ZOR then only the fish in the zone of orientation (ZOO) are considered in the update of the heading. Modelfish in the zone of attraction (ZOA) are only considered if the ZOR and ZOO are empty. We note here the similarity of this configuration with Brooks’ subsumption architecture (Brooks, 1999).

The above swarming systems seem very different but they have the same foundations: they are built on simple rules, and the complex behaviour they exhibit is a by product of these, not explicitly stated by them; they involve extensive interaction between individual members of the swarm using observed information about the other individuals; and the behaviour is highly dependent on the parameters used.

Genetic algorithms (GAs) use emulations of evolution in order to solve computational problems which may be difficult to solve using more conventional techniques Goldberg (1989). The main components are a genome which specifies how the individual is represented in the algorithm, a phenotype which is how the genotype maps onto real world attributes, a fitness function which states how good each individual is and functions to perform mutation and crossover

to produce offspring. Crossover focuses the search on good values by taking two parents and recombining their genomes to create a child; mutation adds a random element to the children which increases variety in the population.

Some work has been done to use genetic algorithms to modify the behaviour of swarms. Conley (2005) used a genetic algorithm to tune a particle swarm optimisation (PSO) search. Geoboids (Macgill and Openshaw, 1998) was the basis of this work, where the swarm moves over a landscape looking for clusters of points, returning the locations of all the clusters it thinks it has found. Conley used a hierarchical fitness function and tournament selection to avoid having to assign each individual an absolute fitness value. Instead, two individuals are compared using a series of criteria, if one wins out on a test then the function stops otherwise the next test is carried out. The tests compare: the number of clusters found; the ratio of distinct clusters to total clusters; the number of dead boids (those which are in empty regions of the dataset when the algorithm finishes); the number of comparisons made (adjusted for the size of the flock) and finally the amount of the dataset searched.

Kwong and Jacob (2003) used genetic algorithms to change many parameters used by a swarm in order to induce “desirable behaviour”, that is swarms which moved in a certain way. They had no automatic fitness function to guide the GA and instead assigned a fitness to each swarm created by the GA by eye, based upon observing its behaviour as it ran. This type of fitness measure could easily be adapted for our work, but it is prone to human error when the differences between individuals are subtle (although some research has been done on methods to improve this (Khemka et al., 2009)).

(Stoops et al., 2010) examined the rules that swarms adhere to by using data mining and rule classification algorithms. A basic experiment into this issue involved running a swarm simulation based on Reynolds’ boids and recording data related to the individual’s movement (the time, position, heading, speed and separation and detection radii). A rule classifier was run on this data to create a series of potential rules. These were then tested by running the boids again but with these rules instead of the defaults. This run was compared to the original and used to modify the rules.

## Methodology

The experiments we report here used a two-dimensional version of the Couzin model (Couzin et al., 2002; Wood and Ackland, 2007) of shoaling behaviour as the basis of simulated fish movement. Like many of the other proposed models it establishes zones of perception around each fish (see figure 1). The location of neighbours within these zones governs the fish’s future movement (being attracted to neighbours in the ZOA, oriented with neighbours in the ZOO, and repelled by those in the ZOR). Our experiments were designed to determine whether it was possible to learn the pa-

parameters of a swarm of modelfish running a known model by adding an agent (robofish) to it and monitoring its behaviour relative to the modelfish. As the robofish behaviour is specified via the same underlying model as the swarm, it follows that if the correct parameters for the robofish model can be set appropriately then its behaviour should be indistinguishable from that of the modelfish. The parameters used by each robofish were encoded in the chromosome of a genetic algorithm. The GA requires a fitness function to evaluate the robofish after a run, which will summarise the robofish's interaction with the modelfish into a single numerical value. The aim of the fitness function is to accurately represent how similar the behaviour of the robofish is to the modelfish. We hypothesised that this should allow the GA to converge on the parameters used by the modelfish, under the central assumption in this work that similar parameters alone will allow a robofish to exhibit behaviour identical to a modelfish.

The motivation for the design of our fitness function was to determine if the robofish was interacting with the modelfish or travelling the arena independently. Accordingly, the experiments used the average Euclidean distance from the robofish to each of the modelfish as the fitness function. This strategy is based on two assumptions: (1), that similar, but not identical behaviour will allow the robofish to interact with the swarm (smooth fitness landscape); (2) that only one configuration of the swarm model will induce this behaviour (no local optima). It is clear that the fitness function needs to be appropriate to the learning task at hand – the parameters of the model are assumed to be impossible to estimate directly, so we can only use the emergent behaviour of the modelfish swarm as the basis for our fitness function. Since this is such an important issue, we took steps to evaluate the fitness measure with respect to the model parameters in the GA.

## Experiments

A single robofish was used in each simulation to ensure that the robofish's behaviour was determined only by its internal model and interaction with the modelfish. If more were used there would be potential for the robofish to interact each other and produce behaviour that was not programmed into the modelfish. In these experiments the only parameters to be changed are the sizes of the three zone widths ZOA, ZOR and ZOO. These were chosen as they control the main aspects of the fish's behaviour.

The parameters used by the model fish in the swarm was set throughout the experiments with parameters used in Wood's original experiment (Wood and Ackland, 2007) (see table 1). This seemed to give an aesthetically realistic swarm whose behaviour often transitioned between different types (Kwong and Jacob, 2003). This would prevent the evolution of robofish who could only swim like the others in certain swarm configurations. With the robofish sharing the same model as the modelfish, the chance of this happening

Parameter	Value
ZOR	1
ZOO	12
ZOA	13
Velocity	2.25
Blind spot	90°
World size	240
Number of modelfish	99
Number of robofish	1
Warmup time	5000
Sample time	5000
Samples	100
Mutation rate	5%
Crossover rate	90%

Table 1: Default values used throughout the experiments

is small. Not much can be done about this before the experiments are run as the behaviour created by adding a robofish to the swarm was unknown *a priori*.

A problem with using these parameters is the repulsion radius ZOR=1. Since setting it to 0 effectively removes this type of perception for the robofish, it is only possible to test what happens when the robofish's ZOR is either larger than the modelfish's or absent (the radius is an integer value). This does place a limitation on the conclusions that can be drawn from the experiments but to change these values would require additional experimentation to find other configurations which produce similar behaviour in the swarm.

The arena dimensions were also the same size as in Wood's experiment: a square 240x240. This means that each modelfish can perceive around 3.7% of the world. For all the experiments in this section the number of fish in each trial (inclusive of the robofish) was set to 100. Preliminary experiments indicated that smaller numbers of fish had a tendency to form multiple groups which either never coalesce into a single swarm or do so only after a very long time interval. It also appears that it is easier for modelfish in a small swarm to escape a group altogether and swim by themselves.

To avoid any bias in the initialisation process, the swarm was run for 5000 time steps before 5000 time steps of monitored behaviour. Here the state of the system was sampled every 50 time steps. This regime reflects the dynamic nature of swarms and reduces the risk of rare, unrepresentative states distorting the fitness estimation. A robofish has to perform well for the entire run to obtain a good fitness score. A robofish which "loses" the swarm or keeps its distance at times will score poorly. In addition, each configuration was performed 5 times to further reduce the effect of the variability in runs on the analysis. This means that each robofish configuration was sampled a total of 500 times.



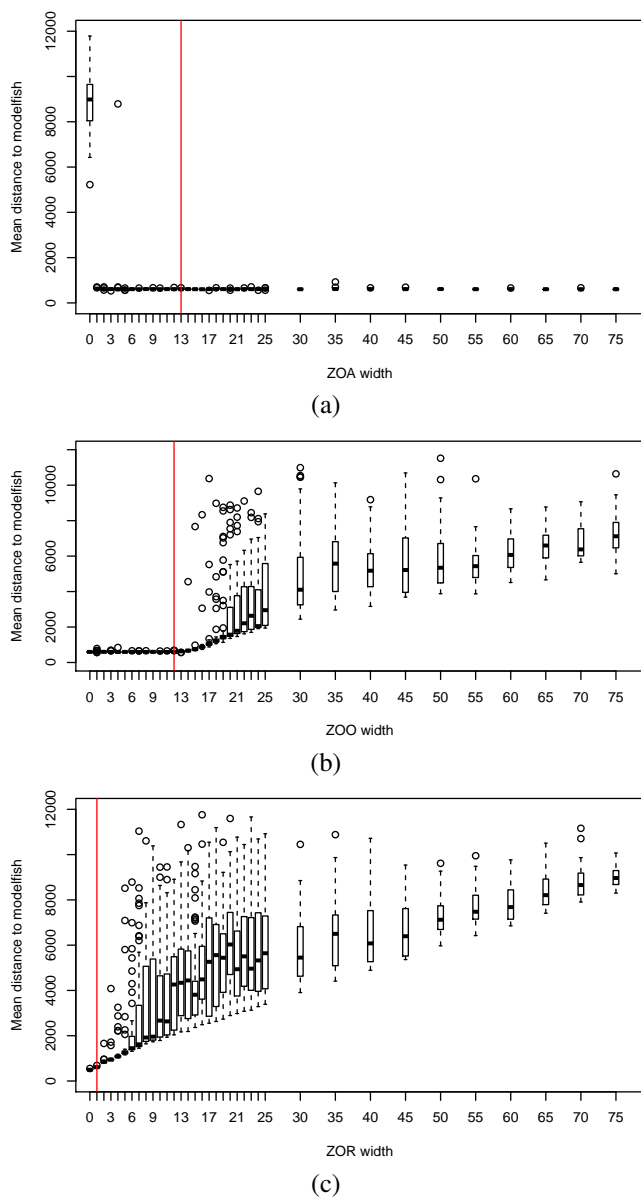


Figure 2: Effect on fitness of changes in widths of (a) Zone of Attraction, (b) Zone of Orientation, and (c) Zone of Repulsion for the robofish. The vertical red line on each plot indicates the target width value used by the modelfish.

### Fitness of zone widths

Our first experiment explored the effect of changing the value of individual zone widths on the fitness of the robofish. The position of the zones relative to the fish are interdependent, as shown in figure 1. Changing one of the widths whilst keeping the other two fixed gives us a clearer understanding of the contribution of the zone width to the emergent shoal dynamics.

Each of the three zone widths were changed from zero

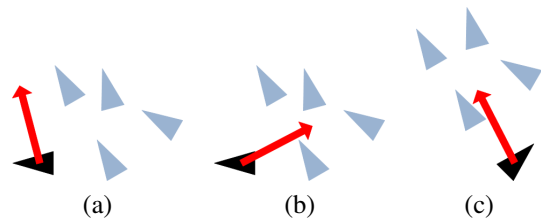


Figure 3: Robofish movement (a) at time  $t$  with a ZOO, (b) at time  $t$  without a ZOO, and (c) at time  $t+1$  without a ZOO. In both cases, the robofish maintains contact with the swarm.

(eliminating the behaviour completely) to 25 (roughly double the modelfish values for ZOO and ZOA) in intervals of 1 to examine how the fitness varies around the values for the parameters actually used by the other fish. The tests were then extended using values from 30 to 75 (roughly six times the modelfish value) in intervals of 5 to explore more dissimilar configurations. Each configuration was tested over 50 trials to allow an accurate representation of the fitness of that value. Figure 2 shows the results of these experiments.

**Zone of Attraction** From figure 2(a) it appears that varying the ZOA has very little effect on the fitness of the robofish. However, when there is no ZOA (i.e. the width = 0), the average distance to the swarm is high, with large variance. The lack of any compulsion to swim towards other fish leads to the robofish being lost from the modelfish swarm. If the ZOA is present, no matter what the size, the robofish performs very well with a low average distance to the modelfish. It would be expected that a high ZOA would allow the robofish to find the swarm more easily if it became separated from the group, but the plot indicates that even a small ZOA with a width of 1 is effective in maintaining the robofish's contact with the swarm. This suggests that the ZOA has a role in limiting the chance of *escape* from the swarm, rather than directing behaviour to *seek* the swarm when swimming alone. Where a ZOA is present in a trial, the swarm has usually formed and contains the robofish by the time fitness measurements commence.

Fish in our model move at 2.25 units per time step. A fish on the margins of the swarm might only have contact with the swarm at the very perimeter of its zone of orientation. There is a small chance that this contact would be lost, and the fish might then escape the swarm if there was no zone of attraction at all. However a very small ZOA would be sufficient to induce the fish to move back towards the swarm should this occur, since those fish which were in the ZOO of the fish in the previous time-step will either still be in the ZOO or have moved to the ZOA – it is very unlikely that they will be completely out of range.

Note also that our fitness measure only samples the position of the robofish after a period of 5000 time steps. If we began to measure fitness from the initialisation period, we

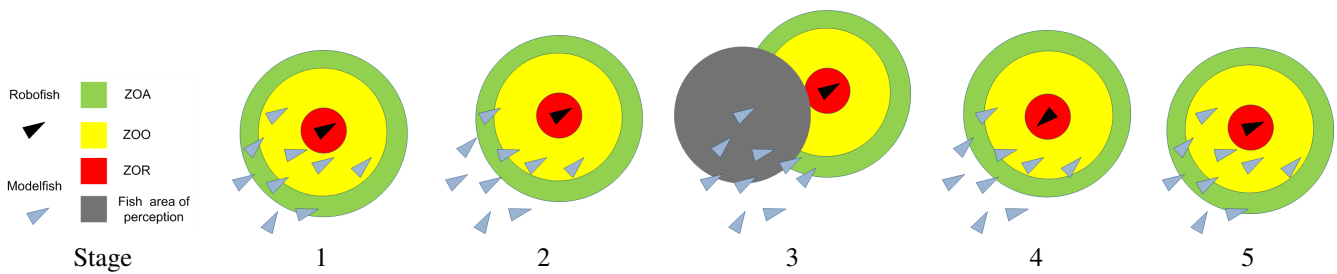


Figure 4: Effects of an individual's large ZOO on shoaling interactions. Stage 1 - The robofish is at the head of the shoal. Stage 2 - The large ZOO allows the Robofish to pull ahead. Stage 3 - The swarm is entirely in the Robofish's ZOA but the robofish cannot be seen by the swarm. Stage 4 - The robofish swims back towards the shoal. Stage 5 - The robofish has rejoined the shoal

may see the effects of varying the ZOA during the process of swarm formation.

**Zone of Orientation** As shown in figure 2(b), the ZOO has a much clearer effect on the fitness of the robofish. At low values for ZOO the robofish performs very well. It is possible that the interplay between ZOA and ZOO (described above) is the reason for this. The behaviour is illustrated in figure 3, where the arrow indicates the movement vector for the robofish in the next time step. In the model at time step  $t + 1$  the ZOA allows the fish to move towards its neighbours, ZOO in the same direction of them. If ZOO is disabled the ZOA will still move towards where the neighbours are which is the same as moving in the direction they were facing at time  $t$ . As the time steps in the simulation are very small and the velocities of the fish small compared to their range of perception (a fish can move a maximum of 2.25 units per time step whereas its radius of perception (ZOA + ZOO + ZOR) is 26 units) the difference between moving towards the heading of the neighbours at time  $t$  and  $t + 1$  is very small resulting in almost identical behaviour.

The mean distance between the robofish and the model fish increases when the robofish ZOO width rises above the ZOO width of the model fish. Between 14 and 19 a slow phase change occurs from a fairly uniform fitness to a more variable, but generally less fit behaviour afterwards. We suggest that this phase change is a by-product of the behaviour a larger ZOO induces. We hypothesise the following behaviour pattern, illustrated in figure 4. The fish in the swarm constantly change relative position in the swarm as they move, due to the stochastic element of the Couzin model. Each fish therefore spends some of the time at the front of the swarm. When the robofish is at the front of the swarm, a situation arises in which the modelfish are in the ZOO of the robofish. At the same time, the robofish is in the ZOA of some of the modelfish, but since other modelfish are in the ZOO of these modelfish, the position of the robofish is ignored by the shoal (stages 1-3 of figure 4). The robofish then changes behaviour, and swims toward the swarm (stages 4-5 of figure 4). As the width of the ZOO increases, there is

an increasing likelihood that the robofish will not successfully rejoin the swarm, since it will be further away from the swarm, and the swarm is more likely to have changed direction.

An increase in the robofish's ZOO increases the distance away from the swarm that it can travel before heading back and allows the robofish to lead again at a point further from the swarm (at lower ZOO values it will rejoin instead). Both these tendencies cause the average distance from the other fish to increase as shown on the graph in figure 2(b). However this does not directly explain the increased variation seen as the ZOO increases. The variation is due to the robofish losing the swarm. The conditions for this seem to be when the robofish has left the sight of the swarm but both the swarm and robofish are moving in almost the same direction. A small deviation from these parallel headings can place members of the swarm within the robofish's ZOA in a single time step, resulting in a situation where the robofish follows the swarm but does not influence it. As the ZOO of the robofish increases, the chance of this happening increases, and small changes in direction cause the robofish to lose sight of the swarm more often. We also postulate an increase in the ability of the robofish to find the swarm again (due to the increased area of the world the robofish can now see). Overall then, as the chances of losing the swarm increase so does the chance of the robofish finding it again, resulting in a lower jump in penalty for losing the swarm. The highest numbers tested show the range of scores decrease again whilst the mean climbs further. At this point the robofish constantly loses the swarm, resulting in uniform but unfit behaviour.

**Zone of Repulsion** As shown in figure 2(c), the ZOR width seems to have a much stronger and simpler effect on fitness than the other two variables. Simply put, the larger the ZOR the further away the robofish is from the others in the swarm. Although the upwards trend is visible from the start, a phase transition of the kind seen with ZOO is shown from around 3 to 7. Once again this is where the robofish starts to lose the swarm on an increasing number

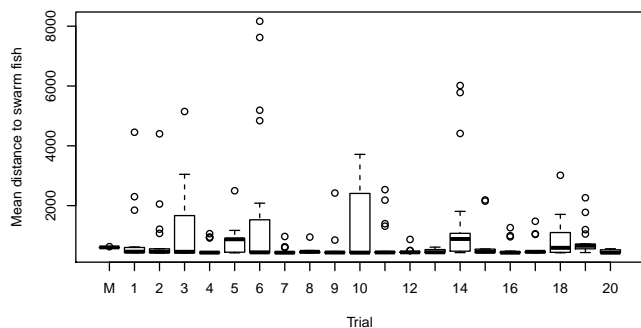


Figure 5: Distribution of fitness at end of 50 generations of 20 trials of the GA. Column 'M' indicates the distribution of fitness for the modelfish

of runs, showing increasing variance in the mean distance to the shoal fish.

### Evolving the model parameters

The previous section demonstrated a relationship between changes in the individual model widths and mean distance to the shoal fish. In this section, we show how a genetic algorithm can be used to find combinations of widths for the ZOA, ZOO and ZOR that minimises the mean distance to the swarm.

We maintained a population of 20 robofish models per generation throughout the trial. This was a compromise between having a large population size (which could explore a sizable portion of the solution space) and minimising the computation time. This is important as each genome must be tested independently since we can only have one robofish evaluation per swarm, meaning that the genetic algorithm will run very slowly. Each robofish configuration was evaluated 5 times.

It was not our intention to evaluate a new configuration of a genetic algorithm. Accordingly, we implemented our genetic algorithm using PyEvolve (Butterfield et al., 2004). Crossover was set to 90%, and mutation (occurring with a 5% chance) changed the Zone widths following a Gaussian distribution with variance of 5% of the current value. The genetic algorithm was run for 50 generations. This was repeated 20 times to estimate the consistency of convergence.

The robofish's genome consisted of the widths of the three zones which were to be modified; the ZOA, ZOO and ZOR. These were stored as integers.

The three zone widths were initialised with random integers in the range [0,75]. The Pyevolve default tournament selection method was chosen to select fit individuals for subsequent generations.

Figure 5 shows the final distribution of mean distance to swarm fish for the robofish population in the final generation of the 20 runs of the genetic algorithm. The mean distance to shoal fish for robofish with the same width values as the

model fish is shown in the column marked 'M'. It can be seen that trials 3, 5, 6, 10, 14 and 18 have not fully converged, but the other fourteen trials show that the genetic algorithm has successfully reduced the mean distance to the swarm fish, as specified by the fitness function. Those trials which did not converge were composed of a mixture of individual models with a combination of low and high mean distances to the modelfish, indicating that it is likely that the runs would eventually converge to low average mean distances across the population if allowed to run for longer.

Note that the mean distance to shoal fish was higher in the control robofish 'M' that used the model zone widths. This reveals an issue with the fitness function - it was designed to evolve a fish that interacted with the shoal, but there is nothing in the fitness function to induce the evolved robofish to *mimic* the behaviour of the model fish. This is why the mean distance is reduced to a minimum, rather than converging on the value that the modelfish parameters generate.

To further illustrate the effect of the fitness function, figure 6 illustrates the change of fitness of the robofish configurations for trial 19, along with corresponding distributions of the three zone widths. In this trial, the genetic algorithm is effective at reducing the mean distance to the swarm. The target widths for the ZOA, ZOO and ZOR are shown as a red line on the bottom three figures.

The width of the ZOA shows the biggest difference between the evolved value and the model value. This is not surprising for two reasons. Firstly, as shown in figure 2(a), the ZOA has little effect on the mean distance to shoal, so it is free to vary. What might explain the drift of ZOA width to such a large value? We assert that a large ZOA is useful in allowing the robofish to find the swarm reliably. If the robofish has not found the swarm by  $t = 5,000$ , then its chance of being selected for the next generation will be reduced.

Both the ZOO and ZOR widths are much more similar to the model values. We claim that the value of the model ZOO is learned effectively by the GA. The ZOR tends to converge to a value of zero, compared with the model value of one. This is a small error, but the consistency of the error leads us to conclude that this value allows the robofish to have a lower mean distance to the modelfish than the modelfish have to each other.

Finally, we note that the ZOO+ZOR widths for the evolved robofish are almost identical to those for the modelfish. This implies that the outer radius of the ZOO, as specified by ZOO+ZOR is a critical parameter in determining the mean distance to other fish.

### Conclusions

The work presented here has started to explore the ability of genetic algorithms to extract important parameters related to a swarm's behaviour. It is only the start of what is possible in this field. Many more aspects can be explored in relation

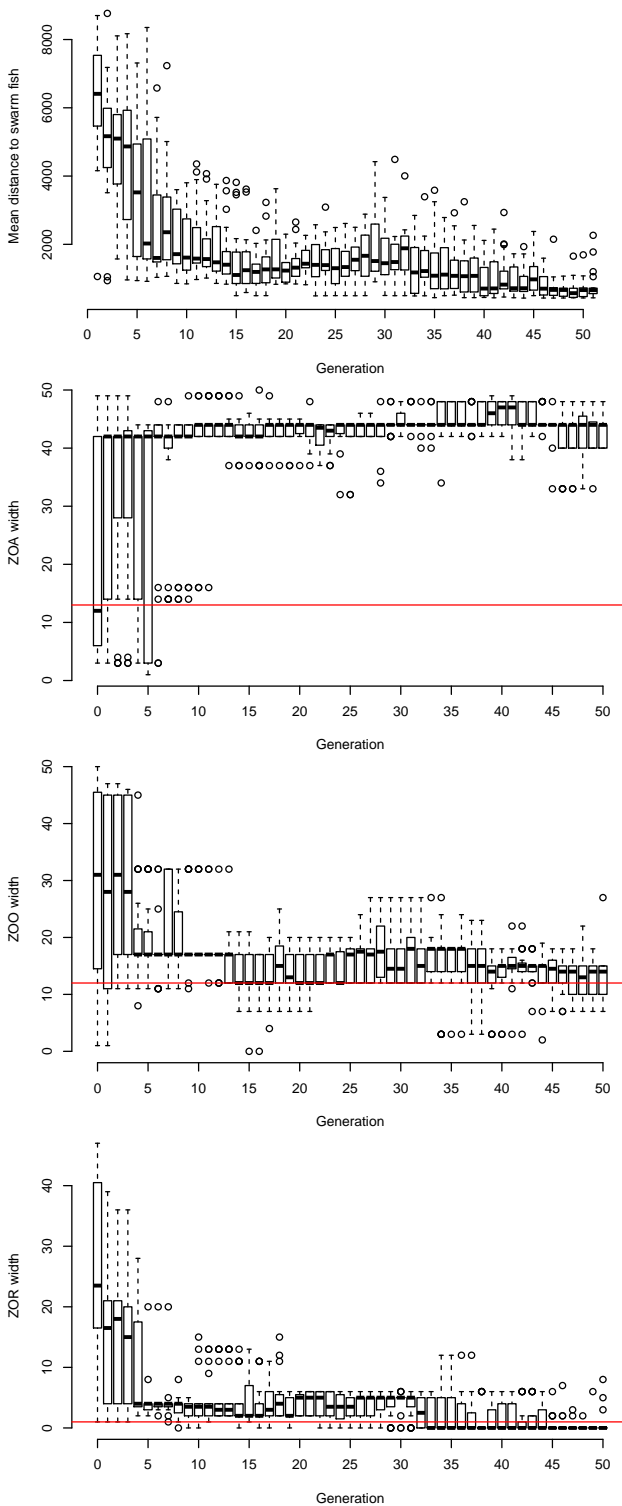


Figure 6: Run 19 of the GA. Plots from top to bottom: Fitness; Convergence to ZOA; Convergence to ZOO; Convergence to ZOR

to the goal of estimating swarm behaviour models via the interaction of configurable agents with swarm agents.

It is well known that the interactions between the zones of behaviour in the Couzin shoaling model lead to an emergent swarm. In this paper, we have investigated ways of recovering the underlying model parameters of the swarm indirectly, via the interactions of an individual with the swarm. The long-term goal of this work is to fit models to observations of shoaling of wild animals. Our initial trials, varying only one of the zone widths whilst holding the others constant, revealed the following observations:

- A ZOA is needed to produce a swarm (i.e. the ZOA width must be greater than zero), but the size of the ZOA makes no difference once the swarm is formed.
- The larger the ZOO the wider the swarm distribution, since individuals can influence the direction of the swarm whilst remaining relatively widely dispersed.
- The ZOR is not necessary for a swarm to evolve. It merely controls the minimum distance between model fish. However, in physically embodied experiments (using robots or fish), it is clear that a ZOR is necessary to reduce the chance of collisions.

The average Euclidean distance to the modelfish was used as the fitness function, which proved effective in evolving individuals which were similar to the modelfish in their behaviour, and had similar values for ZOO and ZOR. The main reason for the difference in these parameters was the simplicity of the fitness measure, which only used the distances between the robofish and the modelfish and made no reference to the distance of modelfish to *each other*. If this technique were to be used with real fish, it would be possible to do this using computer vision techniques, although issues with sampling time might arise.

We suggest that if the measure was changed to use the distance the modelfish are from each other as the target distance rather than simply trying to minimise the mean distance to modelfish, then a more accurate estimation of ZOO and ZOR could be expected. This formulation would penalise robofish that swim too closely to the modelfish as well as those which swim too far away from them. In addition, the ZOA was not well estimated in the framework we devised, but we suggest that this too could be addressed by basing a fitness measure that included the period of shoal formation, rather than basing fitness solely on measures of the swarm after it has formed. Other measures such as the spatial point process measure  $C$  (Getis and Boots, 1978), which compares an individual's distance to its nearest neighbours against a set of randomly determined points in the world space, could also prove to be effective in developing a fitness function that could recover the model parameters.

In addition to a more sophisticated fitness function, other work could focus on exploring the model more fully. The

results gathered from these experiments could vary greatly depending upon the number of robofish used, the time at which the robofish are introduced (whether a swarm has already formed at that point) or the total size of the swarm. These variables would change the interactions between fish and robofish in different ways, potentially producing wildly divergent emergent effects. The results obtained by these experiments would allow swarming behaviour to be understood more fully, potentially providing valuable insights into the way swarm models are designed and configured.

Another clear area of further work is to look into whether this process can be applied to different models of swarming behaviour. There is little reason to suspect it would fail with the similar models from Aoki and Reynolds (although if this were the case it would raise interesting questions about the nature of these models and the extent of their differences), but adapting it to work on more varied models such as ants, termite mounds or wasp nests may prove more challenging and explore the process of extracting parameters more fully (as well as exploring the individual models being tested).

Of course if these methods are ever used on real life swarms it is unlikely that they will correspond exactly to the models proposed so far. By incorporating the ability for the algorithm to modify the model itself (rather than just the parameters of the model), the process could become far more robust at finding an accurate model of behaviour. This could be done by either switching from a genetic algorithm to genetic programming or by changing to genome representation which allows the model to be altered (such as changing the meaning of zones and limiting the number of neighbours looked amongst others). The increase in flexibility this would produce would allow the process to be applied to far more problems, perhaps allowing novel swarm models to emerge via genetic processes.

## Acknowledgments

The authors thank A. Jamie Wood for provision of source code that formed the basis of these experiments. Simon Hickinbotham is funded by the Plazzmid project, EPSRC grant EP/F031033/1.

## References

- I. Aoki. A simulation study on the schooling mechanism in fish. *Bulletin of the Japanese Society of Scientific Fisheries*, 48(8):1081–1088, 1982.
- R.A. Brooks. *Cambrian Intelligence*. MIT Press, 1999.
- A. Butterfield, V. Vedagiri, E. Lang, C. Lawrence, M. Wakefield, A. Isaev, and G. Huttley. PyEvolve: a toolkit for statistical modelling of molecular evolution. *BMC Bioinformatics*, 5(1):1+, January 2004.
- R. D. Coates and S. J. Hickinbotham. Estimating swarm parameters by evolutionary learning. In *GECCO 2011*. ACM, 2011.
- J. Conley. Evolving boids: Using a genetic algorithm to develop boid behaviors. In *GeoComputation*, 2005.
- I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks. Collective memory and spatial sorting in animal groups. *J. Theor. Biol.*, 218:1–11, 2002.
- J.J. Faria, J.R.G. Dyer, R.O. Clement, I.D. Couzin, N. Holt, A.J.W. Ward, D. Waters, and J. Krause. A novel method for investigating the collective behaviour of fish: introducing 'robofish'. *Behav. Ecol. and Sociobiol.*, 64(8): 1211–1218, 2010.
- A. Getis and B. N. Boots. *Models of spatial processes : an approach to the study of point, line, and area patterns*. Cambridge University Press, 1978.
- D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., 1989.
- J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *ICNN*, pages 1942–1948. IEEE, 1995.
- N. Khemka, G. Hushlak, and C. Jacob. Interactive evolutionary evaluation through spatial partitioning of fitness zones. In *EvoWorkshops '09*, pages 432–441. Springer-Verlag, 2009.
- H. Kwong and C. Jacob. Evolutionary exploration of dynamic swarm behaviour. In *CEC '03*, pages 367–374. IEEE, 2003.
- J. Macgill and S. Openshaw. The use of flocks to drive a geographic analysis machine. In *GeoComputation*, 1998.
- C. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Comp. Graphics*, 21(4), 1987.
- D. Stoops, H. Wang, G. Moore, and Y. Bi. Behavioural rule discovery from swarm systems. In *KSEM*, volume 6291 of *LNCIS*, pages 506–517. Springer, 2010.
- A. J. J. Wood and G. J. J. Ackland. Evolving the selfish herd: emergence of distinct aggregating strategies in an individual-based model. *Proc. Roy. Soc. B*, 274:1637–1642, May 2007.

# WASPS: A Weight-Allocated Social Pressure System for the Emergence of Agent Specialization

Denton Cockburn<sup>1</sup> and Ziad Kobti<sup>1</sup>

<sup>1</sup> University of Windsor, School of Computer Science, Windsor, Ontario, Canada  
cockbu3@uwindsor.ca; kobti@uwindsor.ca

## Abstract

Division of labour, or specialization, is common in many types of insect colonies. It emerges in some of these societies as a result of age polyethism, whereby the division of labour is tied to the age of the individuals. One known method that explains this is social inhibition. Individuals release pheromones when they interact with other agents. The strength of their pheromones is tied to their age. These pheromones inhibit the desire of other agents to perform the same task. Using social inhibition, individual agents can be allocated among the available tasks to be performed related to the colony. We apply a variation of this approach to the problem domain where agents can divide their time among multiple tasks. While age is not a factor, agents differ in their skill at performing each task. We create a weight-allocated social inhibition approach whereby more skilled agents inhibit the desire of less skilled agents to perform a task. We are able to see that this approach drives agents toward tasks where they have comparative advantages. This leads to an increase in the division of labour within the population. While inspired by social insects, this approach is easily applicable to agents in other domains.

## Introduction

Specialization is where individuals produce goods and services beyond local or personal need, depending upon other individuals to supply other needed goods. There are many varying definitions of specialization, with most taken from the archaeological, biological and economic fields. One definition from archaeology is that specialization is “the production of substantial quantities of goods and services well beyond local or personal need, and whose production is generally organized, standardized and carried out by persons freed in part from subsistence pursuits” (Arnold and Munns, 1994). By choosing to specialize, specialists must obtain some or all of their subsistence goods through exchange with others (Evans, 1978). There are varying levels of specialization, ranging from being able to sustain oneself, while simultaneously producing goods for the consumption of others, to complete dependency upon exchange with others for subsistence goods. Dependence upon others for subsistence was viewed by Childe as the essence of economic specialization (Childe, 1951).

Specialization allows individuals to maximize their productivity by exploiting their environment (Murciano, 1997), and occurs because entities belong to a community of mutual interest, cooperating to serve that mutual self-interest (Spencer et al., 1998). Specialization may be assigned, as in caste systems, or chosen by an individual driven by varying means, including genetic, social and economic. Another term for specialization is division of labour, which is defined by Hollbloder as “...when individuals can be turned into specialized working machines, an intricate division of labour can be achieved and a complicated social organization becomes attainable even with relatively simple repertory of individual behaviour” (Hölldobler and Wilson, 1990).

There are both internal and external factors that influence an individual’s choice of specialization (Beshers and Fewell, 2001). Internal factors include genetic, neural, hormonal and experience elements. External factors include economic factors such as demand (stimulus) and social influences (Julian, 1999; O’Donnell, 1996; Robinson et al., 1989). It seems that no single behavioural model may fully explain division of labour in complex systems (Traniello and Rosen-gaus, 1997). Different models and approaches have different assumptions, which makes it particularly difficult to compare the effects of factors across different approaches.

The study of specialization is important to several fields. For instance, archaeologists study specialization to understand the changes in societies as a result of the emergence of specialization. It also gives insight into why individuals would choose to produce certain goods over others. From the biological perspective, specialization helps to explain the behaviour of biological creatures such as ants, wasps and bees (Larsen, 2001; Page et al., 1998; O’Donnell, 1996), which have been empirically shown to specialize based on tasks. Economically, specialization is studied to understand its effect upon a society’s economy. It further serves to study how a market may grow or contract based on the specializations present, as these specializations lead to increases in the productivity of market systems (Murciano, 1997). Allyn Abbot Young points out that a productive individual increases the supply of certain commodities, while simultane-

ously increasing the demand for others (Young, 1928). In spite of its role in economics and biology, little is known of the origins and causes of specialization and exchange (Beaudreau, 2003).

In this paper we focus on the social approaches to artificial agent specialization. Here we define an agent as an autonomous social party that can perform several tasks with varying levels of skill. Being social, these parties can also be influenced by their peers across their social networks. It is our hypothesis that competition will drive agents to allocate more of their resources to produce goods with which they possess a comparative advantage in relation to their competitors. As the primary differentiator of efficiency in our model is skill, it can be assumed that more skilled agents will have a comparative advantage over their less skilled competition. In this individual based model, these self-interested agents will be influenced towards performing tasks that will maximize their own productivity. We believe this approach will lead to significant increases in the overall level of specialization within an agent population. In the next section we introduce the social inhibition model from which this work is primarily inspired. We then describe our generic model that uses weight-based allocations. Finally, an experimental setup is presented and discussed with concluding remarks.

### **Social Inhibition**

There are several social models for the emergence of agent specialization. One such method is social inhibition, which implies that as agents choose their specialization, they notify other agents that they have done so, reducing their desire to also choose this specialization. To put that idea in economic terms would be that choosing a specialization reduces the demand (stimulus) for that specialization. Social inhibition aims to explain concepts such as temporal polyethism, which is division of labour based on age, as a result of the interaction between behavioural development and the inhibitory effects of other workers (Huang and Robinson, 1992; Naug and Gadagkar, 1999; Beshers and Fewell, 2001). Temporal polyethism can also be explained experientially, as older agents would have more experience, and thus more knowledge upon which to base their actions (Ravary et al., 2007). This model is more concerned with the physiology of workers and their interactions. Initially, the model took the form of an activator-inhibitor approach, whereby all agents would eventually mature to perform specific tasks, but inhibitors from current performers of these tasks would slow their activation.

Naug and Gadagkar presented a social inhibition model that aimed to explain the age polyethism in wasp species (Naug and Gadagkar, 1999). Their model was in turn based on the verbal model of Huang and Robinson (Huang and Robinson, 1992). In Naug and Gadagkar's model, each agent has two pods: one that increases its own preference for a task, and another that inhibits the preferences of agents

it interacts with for the same task. Their model claimed that individual specialization is emergent from the increase in activator due to age, as well as the amount of inhibitors exchanged when agents interact. The model assumes that all agents possess the same preference and skill level for task performance, which makes it difficult to adapt to situations such as those we aim to address.

The effect of competition on task specialization was examined in (Merkle and Middendorf, 2004). Competition was shown to lead to the occurrence of specialists as an emergent phenomenon dependent on the size of colonies. Their model was based on a genetic preference model though, whereas our model is based on social interactions. They also studied differing demands for tasks, something which we do not explore here.

Another social interaction model was explored in (Gordon and Trainor, 1992). Agents had an active and inactive state for the four tasks in the model. The agents communicate with each other, giving them some idea of how many other agents are performing the same task. These interactions between agents is designed such that the system will trend toward a stable set-point where there is a balance of active and inactive agents for each task. Like the above mentioned models, they also assume that agents do not possess an innate preference or skill for tasks.

A non-social model that is also relevant is (Lavezzi, 2003). Lavezzi's model shows that the amount of specialization and level of per capita output depends on competition, agent connectivity, agent thresholds, and initial conditions such as number of agents and their connectivity. An agent's potential to choose a specialization is limited by the amount of other agents performing the same task, as well as the stimulus level for that task. Agents of course have to know about the level of competition, or be directly aware of the changing stimulus levels. In either of these two situations, agents are required to have excess knowledge of their economic environment. While non-social, we have found that a lot of the effects claimed by Lavezzi are also evident in our model.

The existing social models have several other shortcomings, several of which we look to address. In these models, agents are only able to perform one task per unit of time. In our model, we aim to deal with situations where agents can divide their time among several tasks. Take for example something like human agents, such as those found in (Kohler et al., 2007), who have several tasks to perform in each year such as farming, hunting, getting water and getting wood.

In the social inhibition model, which is aimed at age based specialization, the social influence of other agents do not directly determine the specializations that others will choose. Tasks must first be ranked in a way related to age, then agents are ranked by fit for those tasks. After that, agents are then assigned based on the number of workers needed for that task. We think that while this may be appropriate in insect colonies, it makes the model difficult to adjust to



agent populations where tasks may not have priorities. In our model, we assume no priority among tasks.

## Approach

Our approach is not aimed at system optimization, whereby the system itself tries to be the most productive possible. Instead, agents should be able to emerge the specializations that they are most suited for in their given environment. We assume the existence of a set of tasks  $T$ . Each element  $t$  in  $T$  is a task that can be performed by an agent. Each agent has a level of skill associated with each task. The skill level may be static, or it may be determined by the agent's previous success at performing the task. This allows for skill levels that may correspond with fitness functions in evolutionary algorithms. This skill level is quantifiable, comparable and monotonic, such that  $sk_a(t) > sk_b(t)$  means that agent  $a$  is more skilled than agent  $b$  at performing task  $t$ . All agents assume they can perform the task perfectly. The level of skill is then reflected in the amount of inhibition that agent then releases when they interact with others. Agents are thus able to determine their true relative skill level through interactions with other agents. The strength of inhibition, which we refer to as the influence rate, depends on each agent.

In our test simulations, we assume that all agents have the same level of influence. This is not required, and it is quite possible for different levels to make sense in a domain. For instance, we can create the effect of age polyethism if we were to have the influence rate grow with age. In that case, to create task prioritization, we can have the level of influence vary by task as well. In addition to skill, agents have to divide their time among tasks. They therefore need to track their allocations, which they do internally. Note that while we refer to time, that is simply one idea of a resource. This model does not require the resource to be time, but it can be money, food, or any other divisible resource. The simulation is composed of a set of interacting agents within a social network that can all perform the same tasks at varying skill levels.

## Problem Description

Given agent  $Ag$ , the set of tasks available to  $Ag$   $T_{Ag}$  and a resource  $R_{Ag}$ , how does an agent allocate its  $R_{Ag}$  among each task  $t$  in  $T_{Ag}$ ? So,  $\sum x_i = S(R_{Ag})$ , where  $i$  is each task in  $T_{Ag}$ ,  $S(R_{Ag})$  refers to the amount of the resource  $R_{Ag}$  available, and  $x_i$  refers to a fraction of  $S(R_{Ag})$ . The problem also involves the following conditions: The problem is continuous over a period of iterations,  $S(R_{Ag})$  changes between iterations and  $x_i$  is allowed to change over iterations.

## Weight-based model for resource allocation among tasks

For each agent  $Ag$ , we propose a set  $ALLOC$ , where  $e_i \in ALLOC \Rightarrow$  there is a task  $i$  in  $T_{Ag}$  and  $e_i$  represents the

weight allocated to task  $i$ . Task weights in  $ALLOC$  are relative, therefore for a given task  $i$  and a resource to be allocated  $R_{Ag}$ , the amount of  $R_{Ag}$  to be allocated to task  $i$  is:  $\frac{e_i}{S(ALLOC)} \times S(R_{Ag})$ , where  $S(ALLOC)$  is the sum of all elements in  $ALLOC$ . We make no assumptions about the initialization of the weights in  $ALLOC$ ; they can be randomly assigned, or initialized by some other method. A task having a weight of 0 will result in the task being allocated none of  $R_{Ag}$ . For simplicity, we will assume  $R$  refers to time for the rest of this paper. We also normalize the weights in  $ALLOC$  such that  $S(ALLOC)$  is always equal to 1.

## Model outline

Agents influence other agents when they interact. In some networks, such as kin network, it can be assumed that they interact with all their neighbours in each time step. The amount of influence is dependent on skill level. The higher the skill level, the higher the level of influence. When an agent interacts with another, it positively reinforces its own behaviour, while also inhibiting the other agent. The amount of self-reinforcement is the same amount that it inhibits the other agents. After all agents have interacted, the agent subtracts the level of inhibition it has received from the level of activation it has provided itself. The agent also self-activates itself, such that an agent that does not interact with any other agents will still change its behaviour. These effects result in the change of the allocation levels for the agent.

## Agent Properties

### Agent Attributes

Each agent has the following attributes:

- An allocation set  $ALLOC = \{ t_i \leftarrow [0,1] \}$ , for all tasks  $i \in T$ , where  $t_i$  is the fraction of time the agent will spend on task  $i$ .
- A skill set  $SKILL = \{ s_i \leftarrow [0,1] \}$ , for all tasks  $i \in T$ , where  $s_i$  is the skill of the agent at performing the task  $i$ . If an agent cannot perform a task  $p$ , then the value of  $s_p$  would be 0. The skill level for a task may be dynamic and updated regularly. The skill value as a function must be monotonic though, such that if agent  $Ag1$  has  $s_i$  0.5 and agent  $Ag2$  has  $s_i$  0.7, then we can say that  $Ag2$  is better than  $Ag1$  at performing task  $i$ .
- A set  $PODS = \{ p_i \}$ , for all tasks  $i \in T$ , where  $p_i$  is a 3-tuple  $(A, SA, I)$ . In this 3-tuple for task  $i$ ,  $A$  represents the activator store for the agent,  $SA$  is the level of self-activation, and  $I$  is the inhibitor store for the agent. The agent will increase the weight of the associated task when  $A+SA > 0$ , and decrease it when  $A+SA < 0$ .

The idea behind self-activation is the inclination of an agent to perform more of the task at which they are best. This value should be large enough that it will allow an isolated



agent to specialize over a long period of time, but it should also be small enough that it doesn't overwhelm the social pressure created by stronger competitors.

### Agent Inhibition

The level of inhibition  $I$  in an agent's pod for a task  $i$  is determined by several factors:

- The skill level of the agent at performing task  $i$ .
- The size of the agent's social neighbourhood.
- The influence rate,  $IR = (0, 1]$ , which is a parameter that determines the strength of an agent's influence. This parameter can be universal, or variable for each agent. It is also possible that the influence rate can be different for each task. We can re-create the effect of polyethism if we were to make  $IR$  dependent upon the age of an agent.

### Agent interaction

When agents  $Ag1$  and  $Ag2$  interact, for each task  $t \in T$ , we obtain the values in  $Ag1$ 's pod  $pt$  for task  $t$ , and  $Ag2$ 's pod  $pt$  for task  $t$ . The value in  $Ag1$ 's  $A$  will be decreased by  $Ag2$ 's  $I$  and vice versa. Each agent will also increase its own  $A$  by its  $I$ . This method allows agents to influence each other only when they interact.

Since agents both exchange inhibition, and inhibition level is tied to skill and influence level, the more skillful and influential agent would have a greater effect on a neighbour than a less skillful and influential competitor. While the influence of the "better" agent would be stronger, the weaker agent would still inhibit the stronger one. It is also possible for agents to be considered to interact on every iteration, in which case agents would inhibit all others in their neighbourhood. It should be noted that the level of self-activation plays no role when agents interact.

### Agent Attribute Updates

During each time period, agents will have performed their tasks based on their allocation weights (ALLOC). If the skill set is dynamic, then it would be updated based on the results of task performance. The influence rate of each agent would also need to be updated. If agents have different influence rates for each task, then the updates would need to be applied for each task.

Agents will then update their allocations based on each task pod. Given a normalized allocation  $ti$  for a task  $i$ , and a pod  $(a, s, x)$  for the same task  $i$ , then  $ti$  will be updated as:

$ti = ti + a + s$ . That means that the amount of self-activator  $s$  will be added to the activator  $a$ , and the sum of that added to the current weight. If an agent was overall more skilled at a task than the other agents it interacted with, then its activator level  $a$  should increase. If it is less skilled overall, then the level should decrease, resulting in a negative value for  $a$ . After all task weights are updated for

an agent, the values are again normalized, resulting in the sum of all weights in the agent's ALLOC being 1.

## Experiments and Results

To measure the level of specialization within a population, we use a measure developed in (Gorelick et al., 2004). The measure quantifies the degree to which agents in a population are specialized. We have each agent record their task allocation amounts. These amounts are then stored in an  $n \times m$  matrix, with  $n$  being the number of agents and  $m$  the number of tasks. We then normalize this matrix such that the sum of all cells is 1. The mutual information and Shannon entropy index (Shannon, 1948) are then calculated for the distribution of individuals across tasks. Finally, dividing the mutual information score by the Shannon entropy score will provide a value between 0 and 1. A score of 0 indicates a population with no specialization, while a score of 1 indicates a fully specialized population (Gorelick et al., 2004).

We test our method across several parameter types. These are: the type of network, the number of tasks, the number of agents, and the influence rate. We test with two network types, small-world networks and random networks. Small-world networks (Milgram, 1967) are networks whereby most nodes are connected by a small degree of separation, with the existence of a power-law structure among many nodes. Two famous examples of a small-world network are the '6-degrees of separation' phenomenon found within the US population (Milgram, 1967) and a similar phenomenon among many sites on the World-Wide Web (Bu and Towsley, 2002). With random networks, each node will just be randomly connected with another node. We use the same amount of total edges in both network types, dependent upon the number of agents.

We tested for 2, 4, 10 and 20 tasks. Most studies involve 2 to 5 possible tasks (Waibel et al., 2006), while some insect colonies have anywhere from 20 to 40 specializations (Beshers and Fewell, 2001). Although we could have tested for more possible tasks, we observed that 20 would be sufficient to demonstrate the process. As for the number of agents, we tested smaller groups of 10, 50 and 100 agents, as well as larger groups involving 500 and 1000 agents. Each agent acts after the previous step for all others, meaning that all agents operate in the same time step. Tasks are all assumed to take the same amount of time to perform. We tested with a variety of influence rates, these being 0.05, 0.1, 0.25 and 0.5. The influence rate was the same for all agents during each run. We used a constant self-activation rate of 0.05. All agents also have the same capacity for task performance, that is to say the same amount of time available to be allocated. We ran each combination of parameters 10 times.

Each agent would be created with random task allocations. Thus for each available task, the agent would assign a percentage of their time to be spent on that task. As the metric developed in (Gorelick et al., 2004) is dependent upon

these task allocations, different populations of agents would necessarily have different initial levels of specialization. As such, it is not possible to compare the initial and ending specialization levels across runs within the same network type, even with the same parameter settings. The initial populations would be the same for different network types when all other parameters are the same. Considering these conditions, we measure the change in the level of specialization over a run. In the tables given, rows represent the number of tasks and columns represent the number of agents. Tables 1 through 4 illustrate a representative sample of our overall results. They report the average division of labour (DOL) and standard deviation with influence rates (IR) of 0.05 and 0.5 for both small-world and random networks. The DOL values are average multiples (DOL at beginning of run / DOL at end of run) of the initial level of population specialization over the 10 runs for each parameter combination. Thus a value of 3.3 indicates that there was a 230% increase in the level of specialization. For brevity, the results of other influence rates are not shown.

The level of specialization increased in all 1600 runs that we simulated. In our small-world networks, the average result was a multiple 3.2 over the initial values, with a standard deviation of 0.75. With our random networks, the average result was a multiple of 3.9, with a standard deviation of 0.97. We believe that the higher increase in our random networks is due to the higher average number of connections between agents. In small-world networks, several agents have a lot of neighbours while most have only a few. As agents are influenced by interacting with others, having more interactions result in each agent moving toward its optimal state faster. This suggests that increasing the level of connectivity between agents will result in more pronounced increases in specialization.

Our results may be depressed by the emergence of equilibrium states within our populations. This is the case when adding more iterations will not result in any increase in the population's level of specialization. This emergence of equilibrium states is not surprising though as it is predicted in (Young, 1928). As the initial level of specialization is randomly between 0 and 1, it is the case that a population with a high initial level of specialization would not have much room for improvement. We would not expect to see a state of equilibrium if we had used a dynamic society, as new births, deaths, and other state changes would keep the situation in a state of flux (Lavezzi, 2003).

We noticed that in many cases agents would not become fully specialized. This may be in spite of the fact that they may be significantly better at a particular task than all competitors. This is because they would still have some pressure to perform other secondary tasks where they may still have some advantage. This became more pronounced as the number of tasks increase. In such cases, agents may possess comparative advantages in multiple tasks, and thus the moti-

vation to increase their allocation in both. As the allocation system is weighted, the increases in both weights offset each other.

While we did notice that in most cases increasing the level of influence would also result in a higher level of specialization, this does not occur in all cases. In our simulations, the level of specialization would decline in many cases when going from an influence rate of 0.25 to one of 0.5. Because of the different initial populations and specialization levels, we are unable to study the effect of changing agent and task amounts.

## Conclusion and future work

In this paper we presented a new social inhibition model for the emergence of specialization in agent societies. We showed that this model is able to significantly increase the level of specialization in a random population. While several current models deal with domains where agents can only perform one task at a time, our model deals with having agents that have to allocate their time among several tasks. We have shown that when agents are differentiated by skill level, competition and social inhibition can be used to increase division of labour. We found that our agents will increase their allocation of time among tasks for which they possess a comparative advantage over their neighbours. This follows a well established law of economics. Surprisingly, we also found that using our weight based approach, agents will not necessarily specialize on the task they are most efficient at. This is because the change in allocations for multiple tasks may offset each other. The result seems supported by real world experience, where we have yet to see a modern nation completely specialize on one product. Our model is created in a way that makes it applicable to many domains.

We intentionally kept several parameters abstract because we would like to keep the approach general. Many of the parameters used can be changed to accomodate different domains. We also didn't state how it is that agents interact for the same reason. Interaction could be either broadcast, exchanged through the environment, or exchanged through message passing. The meaning of the social network and its connections is also left open intentionally, such that it could represent a wide range of topics, such as a topographical neighbourhood, or even a collaboration network.

We currently do not account for different levels of resource availability. We would like to investigate what changes if any the model would need to work under those conditions. In addition, we assume that demand is always equal to the amount of a resource produced. It would be a good idea to investigate different levels of demand either globally or locally for each task. We would also like to see how the model performs under dynamic environmental conditions. We would like to apply the model in concrete domains such as human society simulations, or even social insect simulations. We believe that this model can encompass

	10	50	100	500	1000
2	$3.3 \pm 1.24$	$2.48 \pm 0.41$	$2.43 \pm 0.29$	$2.28 \pm 0.16$	$2.28 \pm 0.10$
4	$3.46 \pm 0.95$	$2.86 \pm 0.28$	$2.54 \pm 0.24$	$2.53 \pm 0.08$	$2.48 \pm 0.12$
10	$3.07 \pm 0.65$	$2.73 \pm 0.31$	$2.64 \pm 0.11$	$2.69 \pm 0.07$	$2.71 \pm 0.06$
20	$3.36 \pm 0.42$	$3.08 \pm 0.29$	$2.9 \pm 0.26$	$2.92 \pm 0.12$	$2.88 \pm 0.08$

Table 1: Average DOL multiple and standard deviation with IR = 0.05 in small-world networks.

	10	50	100	500	1000
2	$3.82 \pm 2.11$	$2.76 \pm 0.36$	$2.69 \pm 0.35$	$2.48 \pm 0.16$	$2.53 \pm 0.11$
4	$4.39 \pm 1.46$	$3.4 \pm 0.43$	$3.15 \pm 0.18$	$3.09 \pm 0.06$	$3.03 \pm 0.15$
10	$3.74 \pm 0.80$	$3.4 \pm 0.47$	$3.35 \pm 0.16$	$3.43 \pm 0.09$	$3.45 \pm 0.06$
20	$4.54 \pm 1.06$	$3.53 \pm 0.35$	$3.73 \pm 0.32$	$3.72 \pm 0.14$	$3.72 \pm 0.11$

Table 2: Average DOL multiple and standard deviation with IR = 0.5 in small-world networks.

several of the currently existing social interaction models, including the social inhibition model which inspires it. We didn't think it appropriate to compare our approach to the social inhibition approach here though because they have different assumptions.

## References

- Arnold, J. and Munns, A. (1994). Independent or attached specialization: The organization of shell bead production in californica. *Journal of Field Archaeology*, 21(4):473–489.
- Beaudreau, B. (2003). On the origins of large-scale specialization and exchange: A game-theoretical approach. In *The Canadian Economic Theory Conference*, Montreal, Canada.
- Beshers, S. and Fewell, J. (2001). Models of division of labor in social insects. *Annu. Rev. Entomol.*, 46:413–440.
- Bu, T. and Towsley, D. (2002). On distinguishing between internet power law topology generators. In *INFOCOM*.
- Childe, V. G. (1951). *Social Evolution*. Watts, London.
- Evans, R. (1978). *Early Craft Specialization: An Example for the Balkan Chalcolithic*. In *Social Archaeology: Beyond Subsistence and Dating*. Academic Press, New York.
- Gordon, D.M., G. B. and Trainor, L. (1992). A parallel distributed model of the behavior of ant colonies. *Journal of Theoretical Biology*, 156:293–307.
- Gorelick, R., Bertram, S., Killeen, P., and Fewell, J. (2004). Normalized mutual entropy in biology: quantifying division of labor. *American Naturalist*, 164:678–682.
- Hölldobler, B. and Wilson, E. (1990). *The Ants*. Harvard University Press, Cambridge, Massachusetts.
- Huang, Z. and Robinson, G. (1992). Honeybee colony integration: worker-worker interactions mediate hormonally regulated plasticity in division of labor. In *National Academy of Sciences*, volume 89, pages 11726–11729.
- Julian, G. (1999). *Genetic Variation and Task Organization in the Desert Leaf-Cutter Ant, Acromyrmex versicolor*. PhD thesis, Ariz. State Univ., Tempe.
- Kohler, T., Johnson, C. D., Varien, M. D., Ortman, S., Reynolds, R., Kobti, Z., Cowan, J., Kolm, K., Smith, S., and Yap, L. (2007). *The Model-Based Archaeology of Socionatural Systems*, chapter Settlement Ecodynamics in the Prehispanic Central Mesa Verde Region., pages 61–104. SAR Press, Santa Fe, New Mexico.
- Larsen, J. (2001). Specialization and division of labour in distributed autonomous agents. Master's thesis, University of Aarhus.
- Lavezzi, A. M. (2003). Complex dynamics in a simple model of economic specialization. Technical report, University of Pisa.
- Merkle, D. and Middendorf, M. (2004). Dynamic polyethism and competition for task in threshold reinforcement models of social insects. *Adapt. Behav.*, 12:251–262.
- Milgram, S. (1967). The small world problem. *Psychology Today*, pages 60–67.
- Murciano, A. Millán, J. Z. J. (1997). Specialization in multi-agent systems through learning. *Biological Cybernetics*, 76(5):375–82.

	10	50	100	500	1000
2	$4.42 \pm 2.2$	$2.95 \pm 0.5$	$2.95 \pm 0.37$	$2.82 \pm 0.11$	$2.83 \pm 0.14$
4	$3.62 \pm 1.02$	$3.06 \pm 0.33$	$2.93 \pm 0.27$	$2.92 \pm 0.11$	$2.91 \pm 0.09$
10	$2.99 \pm 0.47$	$2.75 \pm 0.24$	$2.88 \pm 0.1$	$2.87 \pm 0.08$	$2.88 \pm 0.05$
20	$2.99 \pm 0.27$	$2.81 \pm 0.16$	$2.83 \pm 0.11$	$2.83 \pm 0.06$	$2.81 \pm 0.04$

Table 3: Average DOL multiple and standard deviation with IR = 0.05 in random networks.

	10	50	100	500	1000
2	$5.34 \pm 2.79$	$3.79 \pm 0.74$	$3.65 \pm 0.43$	$3.57 \pm 0.16$	$3.59 \pm 0.15$
4	$5.04 \pm 1.57$	$4.48 \pm 0.37$	$4.23 \pm 0.29$	$4.34 \pm 0.13$	$4.28 \pm 0.12$
10	$4.98 \pm 0.70$	$4.65 \pm 0.27$	$4.79 \pm 0.2$	$4.81 \pm 0.12$	$4.83 \pm 0.08$
20	$5.25 \pm 0.35$	$4.84 \pm 0.13$	$4.92 \pm 0.19$	$4.93 \pm 0.08$	$4.91 \pm 0.05$

Table 4: Average DOL multiple and standard deviation with IR = 0.5 in random networks.

- Naug, D. and Gadagkar, R. (1999). Flexible division of labor mediated by social interactions in an insect colony—a simulation model. *Journal of Theoretical Biology*, 197(1):123 – 133.
- O'Donnell, S. (1996). Rapid markers suggest genotypic effects on forager specialization in a eusocial wasp. *Behav. Ecol. Sociobiology*, 38:83–88.
- Page, R., Erber, J., and Fondrk, M. (1998). The effect of genotype on response thresholds to sucrose and foraging behavior of honey bees (*Apis mellifera* L.). *J. Comp. Physiol. A*, 182:489–500.
- Ravary, F., Lecoutey, E., Kaminski, G., Châline, N., and Jaisson, P. (2007). Individual experience alone can generate lasting division of labor in ants. *Curr. Biol.*, 17:1308–1312.
- Robinson, G., Page, R., Strambi, C., and Strambi, A. (1989). Hormonal and genetic control of behavioral integration in honey bee colonies. *Science*, 246:109–112.
- Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656.
- Spencer, A. J., Couzin, I. D., and Franks, N. (1998). The dynamics of specialization and generalization within biological populations. *Journal of Complex Systems*, 1(1):115–127.
- Traniello, J. and Rosengaus, R. (1997). Ecology, evolution and division of labour in social insects. *Animal Behaviour*, 53:209–213.
- Waibel, M., Floreano, D., Magnenat, S., and Keller, L. (2006). Division of labour and colony efficiency in social insects: effects of interactions between genetic architecture, colony kin structure and rate of perturbations. In *R Soc Lond B*, number 273, pages 1815–1823.
- Young, A. A. (1928). Increasing returns and economic progress. *The Economic Journal*, 38:527–542.

# How to prevent intolerant agents from high segregation?

Philippe Collard<sup>1,3</sup>, Salma Mesmoudi<sup>2,3</sup>

<sup>1</sup>University Nicesimilarly-Sophia Antipolis - I3S Laboratory (CNRS - UNS)

<sup>2</sup>Laboratoire d'Imagerie Fonctionnelle - UMR S 678 (Inserm - UPMC)

<sup>3</sup>Institut des Sciences Complexes - Paris-Ile-de-France (ISC-PIF)

philippe.collard@unice.fr

## Abstract

In the framework of *Agent-Based Complex Systems* we examine dynamics that lead individuals towards spatial *segregation*. Such systems are constituted of numerous entities, among which local interactions create global patterns which cannot be easily related to the properties of the constituent entities. In the 70's, Thomas C. Schelling showed that an important spatial segregation phenomenon may emerge at the global level, if it is based upon local preferences. Moreover, segregation may occur, even if it does not correspond to agent preferences. In real life preferences regarding segregation are influenced by individual contexts as well as social norms; in this paper we will propose a model which describes the dynamic evolution of individuals tolerance. We will introduce heterogeneity in agents' preferences and allow them to evolve over time. We will show that it is possible to dynamically get a distribution of tolerance over the agents with a low average and in the same time to deeply limit global aggregation. As the Schelling's model showed that individual *tolerance* can nevertheless induce global *aggregation*, this paper takes the opposite view showing that *intolerant* agents can avoid *segregation* in some extent.

## Introduction

In his article Schelling (1971), Thomas Schelling developed a model of segregation and analysed how a simple preference not be a part of a minority in one's neighbourhood, without necessarily favouring dominance of one's own type, can generate small micro-shocks which have drastic consequences at the macro level. Aggregation happens through a chain reaction, even though the agents do not wish such an extreme situation. Agents interact only locally with their neighbours: every one agrees to stay in a neighbourhood with individuals that have the opposite type, only if there are enough individuals with the same type in the vicinity. This proportion is fixed by a threshold, denoted by the *tolerance* ratio.

More generally, the 'micromotives and macrobehaviour' problematic asks the question of the compliance between local micro-motives and the resulting macro-behaviour. Today, as problems become more and more complex, this problematic is more relevant than ever. In the fields of sociology,

economics, ecology, energy, ..., each one has many *a priori* on the global consequence of his own actions. Most often, a person thinks in good faith that his action will produce faithful results for the community. For example, one can think that:

(a) *intolerant* behaviour lead to *high segregation*

(b) *tolerant* behaviour lead to *low segregation*

Let  $i$  (resp.  $\bar{i}$ ) stands for individual intolerance (resp. tolerance) and  $S$  for a high level of global aggregation/segregation. Hypothesis (a) and (b) can be reformulated by the *micro to macro link*  $[i \rightarrow S]$  and  $[\bar{i} \rightarrow \bar{S}]$ . The Schelling's model provides first an example for the expected case  $[i \rightarrow S]$ ; but, as it shows that tolerance can nonetheless induce a significant level of segregation, it provides also an example for the paradoxical link  $[\bar{i} \rightarrow S]$  where the macro-outcome is intuitively inconsistent with the preferences of the agents who generate it.

This paper shows that macro-segregation can be deeply limited despite the presence of intolerant agents; thus, it provides an example for the dual case  $[i \rightarrow \bar{S}]$ . In the model we propose, each agent has his own threshold of tolerance. At each time, for each agent, the tolerance is adapted using some meta-rules. As a consequence, the emergent state of the 'world' results from a spatio-temporal adaptive dynamics. The scientific question addressed in this work is an evolution of the Schelling Model, which consists in considering an adaptive micro level of tolerance and analysing its impact on the segregation phenomenon observed at the macro level.

This article is organized as follows. In section 2, we propose a generic model of satisfaction. Section 3 shows the global behaviour of models using the simple *Eulogy to Fleeing* rule. Section 4 examines the effects of introducing adaptive tolerance thresholds on the nature of frontier between patterns. Section 5 proposes the new model which allows to conciliate local intolerance and a low level of segregation. Finally, future works are listed and conclusions are drawn.

## A generic model of satisfaction

The Schelling's checkerboard model of residential segregation has become one of the most cited and studied models in many domains as economics, sociology, complex systems science,... Páncs and Vriend (2003), Zhang (2004), Gerhold et al. (2008), Banos (2009). It is also one of the predecessor of agent-based computer models Rosser (1999). Taking inspiration from this model, we define a more generic model of satisfaction (GMS).

The GMS is similar to a 2-D cellular automata model: the 'world' includes numerous agents embedded on a toroidal grid. For each agent, the perception is centered on his local neighbourhood only, where the neighbourhood is constituted of the nearest cells surrounding him. We note  $d_i(t)$  the social *degree* of the agent  $a_i$  at time  $t$ , that is the number of agents in its neighbourhood. Since some locations remain empty, the size of the neighbourhood is the maximum number of neighbours an agent can have. There are two types of agents and each agent has its own type. During a run the agent's type cannot change. The satisfaction of one agent is relative to the type of the agents in its own neighbourhood. For convenience we will denote by a color, *yellow* and *green*, the two possible types.  $Y$  (resp.  $G$ ) represents the set of agents in the *yellow* type (resp. *green* type). Thus, the number of agents is  $(\#Y + \#G)$  and at the global level, the basic hypothesis is  $(\#Y = \#G)$ . At each time  $t$ , for each agent  $a_i$ ,  $s_i(t)$  (resp.  $o_i(t)$ ) represents the number of neighbours with the *same* type (resp. *opposite* type), so  $s_i + o_i = d_i$ .

### From Thresholding to Satisfaction

For each agent  $a_i$ , we assume that there is some quantity measured by the variable  $Q_i$  in the range  $[0, 1]$  which depends on  $s_i$  and  $o_i$ . At each time  $t$ , the value *required* $Q_i(t)$  is a number in the range  $[0, 1]$  which denotes the threshold under which the agent is satisfied according to  $Q_i(t)$ . We define the local boolean indicator *satisfied* as:

$$satisfied_i(t) = (Q_i(t) \leq requiredQ_i(t)) \quad (1)$$

Finally, we define the global indicator *satisfactionRatio* in the range  $[0, 1]$  as:

$$satisfactionRatio(t) = \frac{\#\{satisfied_i(t) = true\}}{\#Y + \#G} \quad (2)$$

This is the ratio of satisfied agents at time  $t$ ; if it is equal to 1, then all the agents are satisfied at time  $t$ .

### Local rule

Once the static description of the model is specified, one must add rules that govern the dynamics of agents' movement. At each time, the motives of each agent are driven by its own satisfaction: an unsatisfied agent is motivated to

move away to one vacant location. The gap between micro-motives and macro-behaviours is due to overlapping neighbourhoods: an agent who moves according to its own interest affects not only the neighbourhood it leaves and the one it arrives in, but also affects, in the long run, all the agents. In GMS we do not fix how an agent moves; this will be specified later when the model will be instantiated. One can only say that there are many ways for an unsatisfied agent to move to a vacant place.

### An index to measure the degree of aggregation

To have some insight into the aggregation level, it is necessary to measure the global aggregate over the world. We reformulate measures proposed by Schelling, Carrington and Goffette-Nagot Schelling (1971), Carrington and Troske (1997), Goffette-Nagot et al. (2009). First, we define a global measure of *similarity* as:

$$s(t) = \frac{1}{\#Y + \#G} \sum_i (1 - Q_i(t)) \quad (3)$$

Then, we define the *aggregateIndex* by

$$aggregateIndex = \begin{cases} \frac{s - s_{rand}}{1 - s_{rand}} & \text{if } s \geq s_{rand} \\ \frac{s - s_{rand}}{s_{rand}} & \text{else} \end{cases} \quad (4)$$

where  $s_{rand}$  is the expected value of the measure  $s$  implied by a random allocation of the agents in the world. A null value for this index corresponds to an average random configuration. The maximum value of 1 corresponds to a configuration with two homogeneous patterns only.

### The Schelling's model of segregation

The Schelling's model of segregation is a particular case for the generic model of satisfaction. In the following we are going to indicate its specificities.

**How to compute satisfaction?** In the Schelling model the quantity  $Q_i(t)$  takes into account the proportion of neighbours of the opposite type; it is computed as the ratio between the number of neighbours having the opposite type and the social degree.

$$Q_i(t) = \begin{cases} \frac{o_i(t)}{d_i(t)} & \text{if } d_i(t) \neq 0 \\ 0 & \text{else} \end{cases} \quad (5)$$

For example, if a yellow agent  $a_i$  has three yellow neighbours and two green neighbours,  $Q_i = \frac{2}{5}$ . If there are no neighbours for the agent (i.e. if  $d_i = 0$ ),  $Q_i = 0$ . If all neighbours have the same type (i.e. if  $o_i = 0$  and  $s_i \neq 0$ ),  $Q_i = 0$ . If all the neighbours are in the opposite type (i.e. if  $s_i = 0$  and  $o_i \neq 0$ ),  $Q_i = 1$ . As the initial spatial configuration is randomly chosen, the initial distribution of  $Q_i$  is binomial.

Table 1: Ratio between the number of neighbours of opposite type to the social degree:  $Q_i = \frac{o_i}{o_i + s_i}$ . Coloured values: agent  $a_i$ , will be satisfied if  $Q_i$  is under the *tolerance* threshold 0.37

$\begin{smallmatrix} o \\ s \end{smallmatrix}$	0	1	2	3	4	5	6	7	8
0	0	1	1	1	1	1	1	1	1
1	0	.500	.666	.750	.800	.833	.857	.875	
2	0	.333	.500	.600	.666	.714	.750		
3	0	.250	.400	.500	.571	.625			
4	0	.200	.333	.428	.500				
5	0	.166	.285	.375					
6	0	.142	.250						
7	0	.125							
8	0								

In this model, all the agents have the same threshold of tolerance: it is a constant value (noted *tolerance*) which is fixed before the run. So, at each time  $t$ , for each agent  $a_i$ , equation 1 becomes:

$$satisfied_i(t) = (Q_i(t) \leq tolerance) \quad (6)$$

The agents are said tolerant if the tolerance is greater than 0.5 ( $0.5 \leq tolerance$ ) and intolerant otherwise. We use the *Moore* neighbourhood commonly employed in agent-based models. So the neighbours of an agent are those living in the eight nearest cells surrounding him and the degree  $d_i$  is a number between 0 and 8. For instance, if the *tolerance* threshold is 0.37, one particular agent  $a_i$ , at time  $t$ , will be satisfied if  $Q_i(t)$  is under this value; this happens in the following eighteen cases: ( $o_i = 0$ ), ( $s_i = 2, o_i = 1$ ), ( $s_i = 3, o_i = 1$ ), ( $s_i = 4, o_i = 1$ ), ( $s_i = 5, o_i = 1$ ), ( $s_i = 6, o_i = 1$ ), ( $s_i = 7, o_i = 1$ ), ( $s_i = 4, o_i = 2$ ), ( $s_i = 5, o_i = 2$ ) and ( $s_i = 6, o_i = 2$ ) (see the coloured values on table 1). More, if there are exactly eight neighbours, i.e.  $d_i = 8$ , (see table 1, the diagonal line) such a tolerance means that the agents are intolerant and cannot suffer more than two opposite neighbours.

**How do unsatisfied agents move away ?** In standard Schelling's models agents move only to satisfy their own interest. This requires that agents must be able to access distant information in order to determine whether or not it will be satisfied in a new vacant cell. This kind of behaviour is characteristic among economical agents that seek to maximize their gain. Nonetheless such a behaviour come out of the idea of agents acting approximately rational, rather economically rational in terms of utility and breaks down the *principle of locality* (see Brownlee (2007)).

**Global behaviour** Regarding the micro-macro problem-atic, the Schelling model provides examples for the two

cases:  $[i \rightarrow S]$  and  $[\bar{i} \rightarrow S]$  where  $i$  (resp.  $\bar{i}$ ) stands for individual intolerance (resp. tolerance) and  $S$  stands for a high degree of global segregation. While the first case is the intuitive situation where micro-intolerance induces macro-segregation, the second case is more surprising as it shows that tolerant behaviours can nonetheless induce a global segregation.

## The Schelling Model with the *Eulogy to Fleeing* rule

In standard Schelling's models agents aspire to satisfy their interests in the new places they move in. In this section, we rather assume a reaction from agent without real cognitive abilities expressed by the simple *Eulogy to Fleeing* rule (EF rule).

### The *Eulogy to Fleeing* rule

The *Eulogy to Fleeing* rule is agreeing with the definition of the term *satisficing* proposed by Herbert A. Simon (1956). "Satisficing describes the selection of a good enough solution, the selection of a decision that meets a minimum threshold or aspiration level, the selection of which occurs in the context of incomplete information or limited computation" Brownlee (2007).

The EF rule is defined as follows: *for each unsatisfied agent, a cell is randomly chosen 'all over the world' and the agent moves to this cell if and only if it is vacant*. So the agents may move at random towards a new location according to their preferences by allowing utility-increasing or utility-decreasing actions. Moves do generate new satisfied or unsatisfied agents by a chain reaction until an equilibrium is reached. At a time  $t$ , if all the agents are satisfied, the EF rule has no effect and then such a configuration is a *fixed point* for the dynamics.

This simple rule is more in the spirit of the complex system paradigm, and, as locally there is no seeking for immediate benefits, it is interesting to know its global consequences. Although it is easy to build some particular cases where the EF rule does not converge, in the following simulations this rule leads the system towards an equilibrium. Let's note that although similar rules based on a random choice of vacant locations are already proposed (Edmonds and Hales (2005), Izquierdo et al. (2009)), they do not look completely identical to the EF rule. In particular, with the EF rule, an unsatisfied agent may stay in place for a while if the randomly chosen locations are occupied.

### Simulation and results

In this paper, all the simulations are realized in the *Net-Logo*<sup>1</sup> multiagent programmable modeling environment Pham (2004), Wilensky (1999). For each simulation, the agent's features are updated in an asynchronous way and

<sup>1</sup><http://ccl.northwestern.edu/netlogo/>



the global geographic parameters are fixed. The world is a square of locations horizontally and vertically wrapped. An agent with type 'yellow' (resp. green) is represented by a yellow (resp. green) square. A black square represents a vacant location. A simulation stops at convergence, when all the agents become satisfied.

The world is a grid-square composed of 10000 locations. This size is a good compromise between the necessity to have a large value to avoid small space effects and the convenience to have a small value to achieve short computation time. There are 1000 vacant locations, knowing that the *density rate* is 90% and 4500 agents in each type. We imposed a random initial configuration: in the cases studied below, the value of  $s_{rand}$  is indistinguishable from 0.5; thus initial configuration induces an *aggregateIndex* closed to 0.

We conducted two types of experiment: in the first one, all the agents are *intolerant* and in the second they are *tolerant*.

**Intolerant agents** For this first experiment the *tolerance* is set to 0.37 (see table 1); so all the agents are intolerant. We can see in figure 1 the result of the agents spatio-temporal evolution at the end of a representative run: after 1150 steps all the agents are satisfied (i.e. *satisfactionRatio* = 1), the mean  $Q_i$  over the whole population (noted  $\bar{Q}$ ) is 0.024 and the *aggregateIndex* is 0.957. From 100 independent runs we obtain, a mean of 0.952 (0.0041)<sup>2</sup> for the *aggregateIndex* and 0.024 (0.0022) respectively for  $\bar{Q}$ . We can observe the emergence of large spatial homogeneous patterns. Moreover the borderland between the patterns is almost build with every vacant location (black square). So patterns are isolated by a *no-man's-land* of vacant cells.

**Tolerant agents** Here, the goal is to show that segregation occurs even if no agent strictly prefers this. We set the *tolerance* to 0.63 (see table 1), so all the individuals are tolerant. In particular, if an agent has exactly eight neighbours, it can bear up to five opposite agents in its vicinity. Figure 2 gives an example of the evolution of the agents' locations during a representative run. At the end, after 228 steps, all the agents are satisfied, the mean  $Q_i$  over the whole population is 0.229 and the *aggregateIndex* is 0.548. From 100 independent runs we obtain, a mean of 0.53 (0.0119) for the *aggregateIndex* and 0.233 (0.0094) respectively for  $\bar{Q}$ . While spatial segregation is not an attribute of the rational individuals's behaviour, we can observe the emergence of many segregationist patterns, although they have a smaller size that in the previous case (see figure 1). More, vacant locations are scarce on borderline because with a high tolerance level vacant cells are not requisite to delimit segregationist patterns.

<sup>2</sup>standard deviation is shown in ()

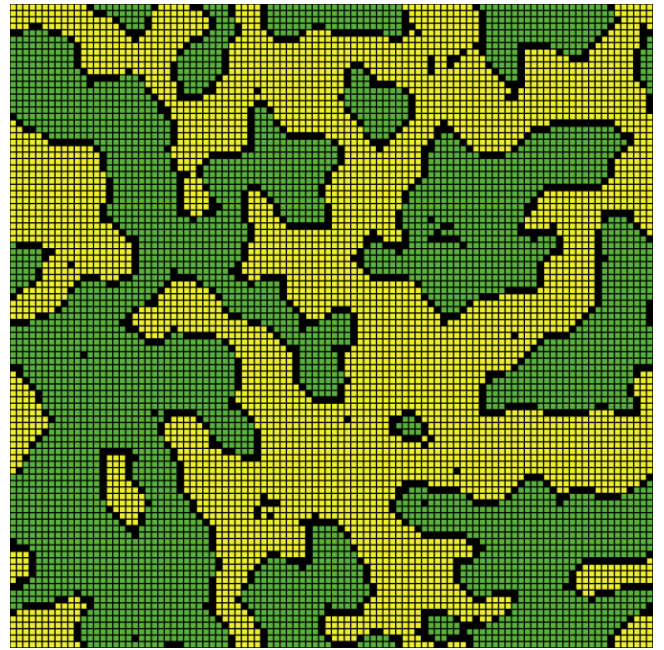


Figure 1: The Eulogy of Fleeing rule: *tolerance* = 0.37  
View at convergence (*ticks* = 1150):  $\bar{Q}$  = 0.024  
*aggregateIndex* = 0.957

## Discussion

In this section, we have shown that in spite of the use of a more simple and realistic local rule, the model produces a comparable global behaviour than the classical Schelling model.

We have shown that both intolerant and tolerant local behaviours lead to the satisfaction of all the agents with the emergence of global segregationist patterns. Moreover, the gap between the *tolerance* and the mean  $Q_i$  over the whole population is surprisingly large at the end of a run. In this way complex dynamics build much more liveable configurations than necessary. With intolerant agents, vacant places are required to form the frontiers and insulate agents in homogeneous patterns. In the next section, we propose to modify the model in order to insulate segregationist patterns without using vacant locations mainly.

## From *no-man's-land* to *mediator-land*

Most often, in real life some individuals are tolerant whereas others are intolerant. In a model, there are two ways to take into account this fact: either fixing a distribution for the tolerance, or dynamically evolving tolerance to 'converge' toward a particular distribution. The first solution requires not only to choose one distribution: *uniform*, *normal*, *poisson*,... but also to fix its parameters: mean and standard deviation.



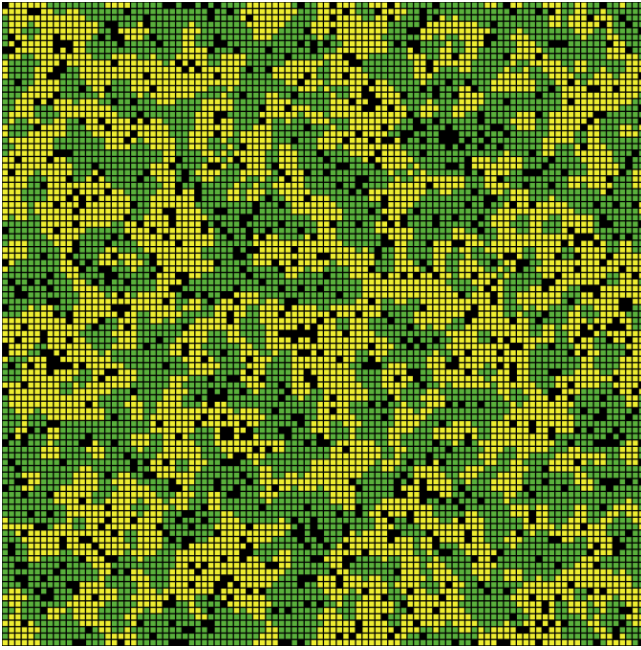


Figure 2: The Eulogy of Fleeing rule:  $tolerance = 0.63$   
View at convergence ( $ticks = 228$ ):  $\bar{Q} = 0.229$   
 $aggregateIndex = 0.548$

### Adaptive local rule

As we have no *a priori* on a target level of tolerance, we choose to start from an intolerant configuration and to apply a local rule to gradually increase the tolerance. For instance, when a person is immersed in an unknown world, his first attempt will be to meet people which look like him; so initially, certainly with many *apriority*, such a person is gregarious or intolerant. Then, if his requirement is too high relatively to the environment, it will be difficult for him to find a fitting place; therefore a natural tendency will be to gradually reduce his stress by decreasing his gregariousness and/or increasing his tolerance.

In this new instance of the GMS, each agent has its own tolerance threshold. Furthermore, each individual threshold may vary over time. So, for each agent  $a_i$ , at each time  $t$ , the *satisfied* indicator (see equation 1) becomes:

$$satisfied_i(t) = (Q_i(t) \leq tolerance_i(t)) \quad (7)$$

Initially, the tolerance of each agent is set to a very small value, therefore an agent is at first radically intolerant and so will be unsatisfied. At each time, for each unsatisfied agent, a cell is randomly chosen 'all over the world' in order to move in if it is vacant, otherwise, i.e. if the cell is already occupied, the agent stays put and adapts its own *tolerance* to the context by increasing its value with a small increment.

### Simulation and results

For each agent, the *tolerance* is initialised to 0.001 and we chose a small increment of 0.003. We can see on figure 3 the spatio-temporal evolution of the agents at the end of a representative run. After 663 steps all the agents are satisfied; the mean *tolerance* over the population is 0.365, the mean  $Q_i$  over the population is 0.049 and the *aggregateIndex* is 0.892. Even if dynamics are more complex than in Schelling's model, we can observe the emergence of spatial homogeneous patterns yet. On 100 runs we obtain, a mean of 0.919 (0.0120) for the *aggregateIndex* and 0.360 (0.0047) respectively for the mean *tolerance*. So, on average, dynamics lead agents to remain intolerant and a high segregation emerges at the global level; once again this is an example for the case  $[i \rightarrow S]$ .

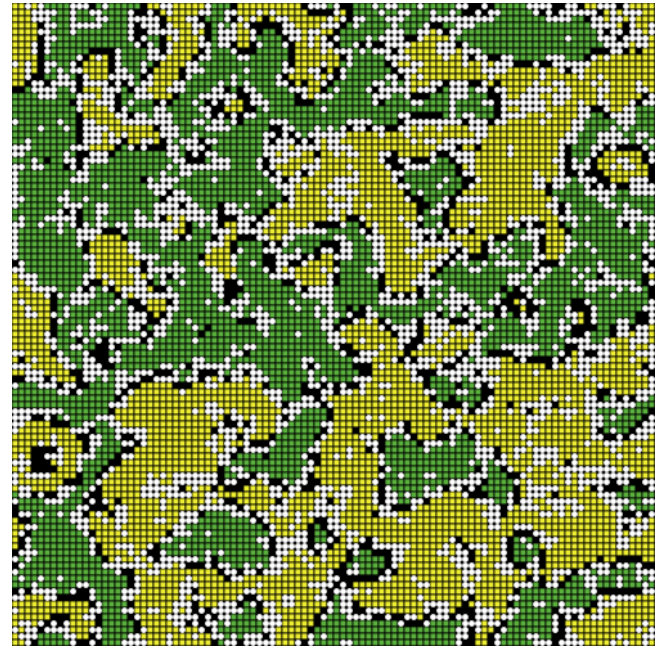


Figure 3: Dynamic tolerance  
View at convergence ( $ticks = 663$ ):  
 $aggregateIndex = 0.892$  mean  $tolerance = 0.365$

We can observe that the frontier between homogeneous patterns is constituted both by vacant cells (black square) and by the most tolerant agents (white circle), i.e. agents with  $tolerance \geq 0.39$ ; therefore, for a significant part, homogeneous regions are isolated by *places for mediation* where opposite agents may co-exist. We can note that there are also tolerant agents outside the *mediator-land*; this corresponds to *scoria*<sup>3</sup> in some areas where former conflicts have led to the local hegemony of one of the two types; thus data collected from the own tolerance of the agents allow to learn

<sup>3</sup>Scoria is the dross that remains after the smelting of metal from an ore

more about the past of the system.

## Discussion

A first result is that dynamics leads the mean tolerance toward a relatively weak value (0.36); as a consequence, when all the agents become satisfied, they remain on average intolerant. The second result is that segregation is still high (0.919). The third result is that in a world where agents are on average intolerant there are some tolerant agents which play a crucial role in the spatial distribution. This can serve as a clue to extend the model toward more *mosaic-like* structure. *Type-mix* would be favoured by the existence of secluded agents amidst individuals having an opposite type. In the present model, this is impossible because agents are not tolerant enough to endure such a situation: we have to enhance the dynamics to allow tolerance to reach high values. On the contrary, the presence of *scoria* shows that one agent with high tolerance may be useful in a moment at a place then becomes superfluous later in the same location; so decrease the tolerance of satisfied agents may help to avoid such 'frozen region'. All this suggest us to manage two antagonist dynamics: increasing and decreasing the tolerance; so, we expect to significantly lower the level of segregation while maintaining a weak mean tolerance.

### How to avoid high segregation ?

In this last section the goal is to respond to the question: *How intolerant agents can become satisfied without the emergence of macro segregation?*

In the new model we propose, there are two antagonist dynamics, the first one increases the tolerance of unsatisfied agents, whereas the second decreases the tolerance of satisfied agents. Initially, the agents have a weak tolerance and are thus radically intolerant and unsatisfied.

- An unsatisfied agent, can either move to a vacant place or else simply increase its tolerance (for details, see the previous section).
- Conversely, for a satisfied agent  $a_i$ , if the difference  $\delta$  between its  $\text{tolerance}_i$  and the value of  $Q_i$  in the place it lives in is too high, its tolerance decreases.

In real life, when a person is no longer confronted with distressing circumstance, his ability to cope later in such a situation is reduced. This phenomenon can be explained by a mechanism of *forgetfulness*. In the model, an agent is satisfied if it is not faced to a large enough number of opposite agents. If over time such a lack of confrontation persists, then the agent gradually reduces his threshold of tolerance.

### Parameter space exploration

There are two main parameters that control the dynamics of tolerance: the amount of increment  $\text{inc}$  and decrement  $\text{dec}$ .

First we conduct a parameter space exploration in order to choose suitable values for the simulation.

In the context of complex systems, most often there are several parameters which together determine the global dynamics. In order to choose values for the parameters used in the simulations, we have first conducted an exploration of the *parameter space*. The objective to minimize both the mean *tolerance* and the global *aggregateIndex* is difficult because when the first one decreases, the second increases and conversely. Therefore, we conduct a tradeoff-analysis to identify compromise for which the two criteria are mutually satisfied in a *Pareto-optimal* sense. This is a typical multi-objective optimisation problem where the optimal solutions correspond to a set of compromises expresses by a *Pareto front* Dyer et al. (1992), Belton and Stewart (2002). In practice, the Pareto front is proposed to a human decision-maker who then chooses a solution according to his expertise.

For all the tests we perform, the parameter  $\delta$  is set to 0.1. We focus our effort on areas that lead to interesting regions where convergence occurs with low *tolerance* and low *segregation*: each test corresponds to one couple ( $\text{inc}$ ,  $\text{dec}$ ) in the range  $[0.025, 0.040] \times [0, 0.030]$ . There are 60 tests and, for each one, results are averaged over 100 runs. Each data point of the scatter plot (see figure 4) corresponds to a couple ( $\text{inc}$ ,  $\text{dec}$ ) and represents both the *aggregateIndex* (y-value) and the mean *tolerance* (x-value) obtained when all the agents are satisfied. We can observe that heightening the parameter  $\text{dec}$  (while  $\text{inc}$  remains constant) pushes the point solution to the left toward the *Pareto front*. Conversely, lowering the parameter  $\text{inc}$  (while  $\text{dec}$  is constant) moves up the point solution on one front. This analysis leads us to choose a particular point on the *Pareto-front* that represents a good compromise between both intolerance and low segregation. To conduct the following simulations, we choose the point corresponding to the parameter values  $\text{inc} = 0.029$  and  $\text{dec} = 0.017$  (See the arrow on figure 4).

## Results

Initially, all the tolerances are set to 0.1. We can observe on figure 5 the spatial configuration at the end of a representative run when all the agents are satisfied: after 513 steps, the mean  $Q_i$  over the population is 0.306, the mean *tolerance* over the population is 0.369 and the *aggregateIndex* is 0.383. On 100 runs, we obtain on average an *aggregateIndex* of 0.388 (0.0110) and a mean *tolerance* of 0.370 (0.0048). The value for the *aggregateIndex* (0.388) has to be compared with the ones obtained with the two previous models (0.957 and 0.919)

The frontier between homogeneous patterns is constituted by the most tolerant agents and there is no *scoria* inside the patterns. One observes that homogeneous areas are infiltrated by many secluded individuals: there are some niches which co-exist within a cohort of unlike agents; this is possible only because loners are very tolerant. In contrast with

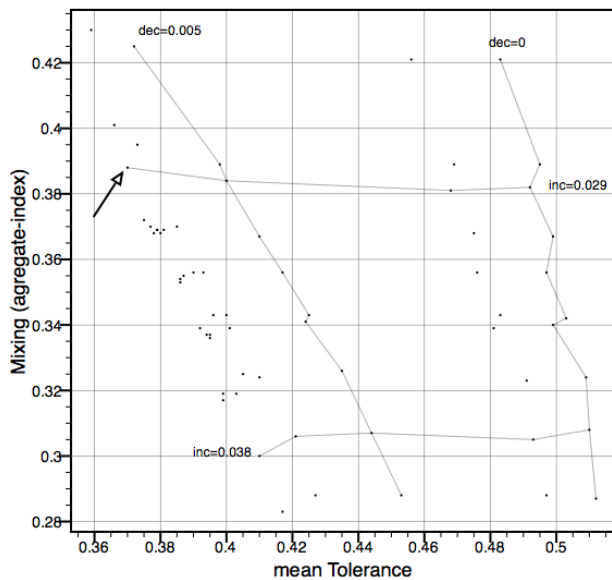


Figure 4: Parameter space analysis  
Tolerance vs. Segregation

the previous models, vacant locations don't play any role in isolating individuals from each other. The most important feature of this model is that it prevents intolerant agents from high segregation. As the Schelling's model provided an example for the case  $[i \rightarrow S]$ , this model exemplify the  $[i \rightarrow \bar{S}]$  micro-macro link.

### Conclusion and future work

In this article, we have proposed to extend the Schelling's model considering that every individual has its own tolerance level. In a first step we have proposed a simple way to locally manage the tolerance; all that gives rise to the emergence of a new kind of border and inner *scoria* both made up of the most tolerant agents. In a second stage, we have introduced new dynamics that consists of combining two antagonist strengths. As a result of this confrontation, the agents are able to reach an equilibrium where they all are satisfied, rather intolerant, but where the aggregation level remains low. As, at our knowledge, there is no prior work on this topic, this result is a significant challenge to the analysis conducted by Schelling: it shows that one can avoid segregation if the tolerance level is adaptive, which is in our opinion a better assumption.

In future work, we will revisit those results by considering situations closer to reality. Beyond a simple world of agents embedded on an homogeneous toroidal-grid, we have to consider different types of network as for example neighbourhoods defined from a scale-free network. We have observed the emergence of very different type of frontiers: *no-man's-land*, *mediator-land* or in some extend *mixing*; thus,

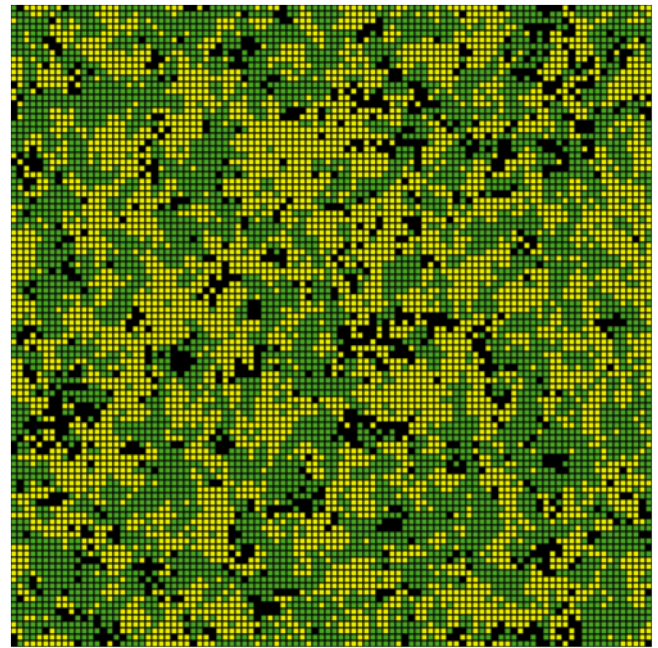


Figure 5: Intolerant agents avoid global segregation  
View at convergence (*ticks* = 513): mean  
*tolerance* = 0.369, *aggregateIndex* = 0.383

it might be interesting to study for a border, its composition, its spatial distribution, its volume, porosity, permeability,... and so to better understand its function: place of exchange and/or medium to isolate.

### Acknowledgements

We thank the referees for useful comments on the manuscript and Touria Ailt Elhadj for her english correction.

### References

- Banos, A. (2009). Exploring network effects in schelling? segregation model. *S4-MODUS workshop Multi-scale interactions between urban forms and processes*.
- Belton, V. and Stewart, T. (2002). *Multiple Criteria Decision Analysis: An Integrated Approach*. Springer-Verlag.
- Brownlee, J. (2007). Satisficing, optimization, and adaptive systems. *CIS Technical Report 070305A*.
- Carrington, W. and Troske, K. (1997). On measuring segregation in samples with small units. *Journal of Business & Economic Statistics*, pages 402–409.
- Dyer, J., Fishburn, P., Steuer, R., Wallenius, J., and Zionts, S. (1992). Multiple criteria decision making, multiattribute utility theory the next ten years. *Management Science*, 38(5):645–654.
- Edmonds, B. and Hales, D. (2005). Computational simulation as theoretical experiment. *Journal of Mathematical Sociology*, 29(3):209–232.

- Gerhold, S., Glebsky, L., Schneider, C., Weiss, H., and Zimmermann, B. (2008). Computing the complexity for schelling segregation models. *Nonlinear Science and Numerical Simulation*, 13 (10):2236–2245.
- Goffette-Nagot, F., Jensen, P., and Grauwin, S. (2009). Dynamic models of residential segregation: Brief review, analytical resolution and study of the introduction of coordination. *HAL-CCSD*.
- Izquierdo, L., Izquierdo, S., and Galan, J. and Santos, J. (2009). Techniques to understand computer simulations: Markov chain analysis. *Journal of Artificial Societies and Social Simulation*, 12(16).
- Pancs, R. and Vriend, N. (2003). Schelling’s spatial proximity model of segregation revisited. *Computing in Economics and Finances*.
- Pham, D. (2004). *From Agent-Based Computational Economics towards Cognitive Economics*. in Bourguin P., Nadal J.P eds : Cognitive Economics: An Interdisciplinary Approach. Springer verlag.
- Rosser, J. B. J. (1999). On the complexities of complex economics dynamics. *Journal Of Economic Perspectives*, 13:169–192.
- Schelling, T. C. (1971). Dynamic models of segregation. *Journal of Mathematical Sociology*, 1:143–186.
- Simon, H. A. (1956). Rational choice and the structure of the environment. *Psychological Review*, 63:129–138.
- Wilensky, U. (1999). Center for connected learning and computer-based modeling. <http://ccl.northwestern.edu/netlogo/>.
- Zhang, J. (2004). A dynamic model of residential segregation. *Journal of Mathematical Sociology*, 28:147–170.

# A Loop Conjecture for Metabolic Closure

D.A. Contreras<sup>1,‡</sup>, U. Pereira<sup>1,2,‡</sup>, V. Hernández<sup>2,‡</sup>, B. Reynaert<sup>2,‡</sup>, J.C. Letelier<sup>2,‡</sup>

<sup>1</sup> Departamento de Física, Facultad de Ciencias, Universidad de Chile, Casilla 653, Santiago, Chile.

<sup>2</sup> Departamento de Biología, Facultad de Ciencias, Universidad de Chile, Casilla 653, Santiago, Chile.  
letelier@uchile.cl

## Abstract

Although in the last few decades a variety of theoretical tools have been developed to better understand living organisms, their impact on experimental research has been rather limited. A common element between these theories is the idea of metabolic closure, i.e., the systems that produce all their metabolites and catalysts. In spite of an increasing consensus on the relevance of closure, a formal and operative definition has remained elusive. In this paper we revisit RAF sets and chemical organization theory and show how these two theories overlap and could help bring forth real world results. We also state a theorem ensuring the presence of a cycle of interdependent catalysts for RAF sets and conjecture that these cycles give stability to the network. This conjecture is illustrated and supported by computer simulations. Unavoidably, our viewpoint introduces the notion of fluxes and thus a temporal dimension to the purely algebraic model of RAF sets. The results of this work show that the incorporation of closure, topological and dynamical tools altogether is a promising path for a deeper understanding of living systems.

## Introduction

In the last thirty years there have been many efforts directed to develop theories to understand biological systems in terms of metabolic closure or, equivalently, systems that produce and maintain themselves. Two crucial models that definitively put metabolic closure at the very center of biological organization are: Autopoiesis, formulated by Maturana and Varela (Maturana and Varela, 1980), and Rosen's ( $M, R$ ) Systems (Rosen, 1958). But these two theoretical studies and similar theories (like the Chemoton or Autocatalytic sets), although very clarifying in basic aspects, have not yet produced technical results that illuminate the daily life of bench biologists involved in experimental research.

In the past (Jaramillo et al., 2010) we have emphasized that a little known formalism called RAF sets (Hordijk and Steel, 2004) is a particularly suited technical tool to understand closure in general and autocatalytic sets in particular. Here we study the relation between RAF sets and the chemical organization theory (COT), which is a theory that

adds to the dynamical aspects by introducing the notion of metabolic fluxes to the purely algebraic vision of RAF sets, an idea deeply embedded in basic metabolic engineering. This is accomplished by expressing the kinetic behaviour of the components (molecules) in terms of a stoichiometric matrix, which then leads directly to the concepts of rates and fluxes, introducing the temporal dimension. This approach can be used to expand the original RAF sets theory, which we consider to be highly valuable for biology, but unfortunately too algebraic to be of use, in particular lacking a way to describe the time behaviour of the systems, which is of most importance in the direction of a more realistic biological context.

Here we will show how notions from chemical kinetics can be fused with RAF sets to search for closure in metabolic networks. Although the results presented here seems, initially, as a mere technicalities without theoretical relevance, they open new research paths as we adjoint highly theoretical notions (RAF set and the metabolic closure) with an accepted used tool to understand metabolism in steady state. In particular we show the logical relation between COT and RAF sets.

## RAF sets and COT in a Nutshell

We now give a brief introduction to the work of Hordijk and Steel (2004), who came up with a formal framework to study a autocatalytic systems. Their main aim appears to have been to develop algorithms with which autocatalytic systems in Kaufmann's sense (1993) could be described and found computationally. They have produced a powerful approach that can be used to analyze a wide variety of systems. Their formalism is based on the following two important sets:  $X$  is the set of molecules involved in metabolism (i.e. metabolites, catalysts or external input material, termed *food set* in the formalism), and  $\mathcal{R}$  is the set of reactions that define the metabolic network. Each reaction  $r$  is represented as a tuple  $(A, B)$ , where  $A, B \subset X$ ,  $A \cap B = \emptyset$ ,  $A$  are the reactants and  $B$  the products of reaction  $r$ .

Further, to formalize the notion of catalysis, a specific set  $C$  (called the set of "catalyzations" by Hordijk and Steel)

<sup>‡</sup>All authors contributed equally



is introduced. Each catalyzation  $c$  is a tuple  $(x, r)$ , where  $x \in X$  is the catalyst and  $r \in \mathcal{R}$  is the reaction catalyzed by  $x$ . Additionally, the subset of molecules that are used but are not produced by metabolism is called food and denoted by  $F$ . Thus, a *catalytic reaction system* over a food source  $F$  is composed by a triplet  $\mathcal{L} = (X, \mathcal{R}, C)$  which defines the universe of molecules ( $X$ ), the reactions occurring among these molecules ( $\mathcal{R}$ ) and the identity of the catalyst involved in each reaction ( $C$ ). Note that this already provides, although at a very simple level, a way to refer to a system, and distinguish the inner and outer components and the transformations that the components undergo.

The following additional functions are defined:  $\rho(r) = A$  and  $\pi(r) = B$ , which return the reactants and the products of any given reaction  $r$ , respectively, and the function  $\text{supp}(r) = \rho(r) \cup \pi(r)$ . With the help of these elementary functions, the same notion can be extended to a *set of reactions*  $\mathcal{R}'$  as  $\rho(\mathcal{R}') = \bigcup_{r \in \mathcal{R}'} \rho(r)$ , where  $\mathcal{R}' \subseteq \mathcal{R}$ . This definition captures the conglomerate of molecules that participate as reactants for a set of reactions. A similar definition holds for  $\pi(\mathcal{R}')$ , the products of a subset of reactions. With these ideas we can define the closure of a subset  $X' \subseteq X$  relative to  $\mathcal{R}' \subseteq \mathcal{R}$  ( $\text{cl}_{\mathcal{R}'}(X')$ ) as the set of reachable molecules that can be synthesized by starting from  $X'$  and iteratively applying all the reactions in  $\mathcal{R}'$ . Note that this definition is of most importance, as it follows that a set of molecules which is closed (i.e. it is equal to its closure) under a set of reactions will not generate any new molecule and thus, conserves its identity. This operation captures the central idea of metabolic closure, which is fundamental for achieving organizational invariance in autopoietic systems. A *catalytic reaction system* is *reflexively autocatalytic* if for each  $r \in \mathcal{R}$  there is an  $x \in \text{supp}(\mathcal{R})$  such that  $(x, r) \in C$ . In other words, every catalyst must be a reactant or product of a reaction in the same system. The system is *F-generated* if every reactant is either produced by the system or incorporated as a food item (i.e. formally  $\rho(\mathcal{R}) \subseteq F \cup \pi(\mathcal{R})$ ). A system that is *reflexively autocatalytic* and *F-generated* is called a *RAF set* (see figure 1).

RAF sets can be understood informally as an interdependent set of biochemical reactions where all of the metabolites, with the exception of the so-called food set, are produced by the collection of reactions  $\mathcal{R}$ . This self generation, a defining feature of autopoietic and  $(M, R)$  systems, is the core of metabolic closure. Thus, RAF sets, autopoietic and  $(M, R)$  systems overlap to a great extent; positioning RAF sets as an operative theory to metabolic closure. The advantage of RAF set formalism is that it is precise enough to be coded in well defined algorithms that exploit its intrinsic recursiveness. To check if a given collection of biochemical reactions is indeed a RAF set, Hordijk and Steel (2004) developed algorithms aimed to analyze the interdependence between a given catalyst and its production pathway.

The chemical organization theory, initially developed by

Dittrich and Di Fenizio (2007), deals with chemical reaction networks. In what is called *static analysis*, the part of this theory that is concerned with the topology of the system, molecules and reactions are defined in a very similar way as in RAF sets. Most notably, both theories share the definition of the closure operator. But while COT makes no explicit mention to catalysts and therefore distances itself from biological systems in which this concept is fundamental, it does incorporate tools to study the dynamical behaviour of chemical reaction networks, thus provides a connection between the structure of a system and the dynamical aspects of it. This is accomplished by first expressing the system in terms of the stoichiometric matrix and associated differential equations.

In COT, it is useful to recognize systems fulfilling certain properties, such as closure. For example, a system is self-sustained if it is able to generate every molecule that is used up. When this topological consideration is transported to the time domain, we can define mass-maintaining systems. A system is said to be mass-maintaining when:

1. All reactions that can be fired by the molecules in the system occur at some positive rate
2. Reactions whose reactants are missing from the system do not occur
3. There is a combination of reaction rates such that all molecules increase or maintain their concentration.

A system which is both closed and mass-maintaining is called an organization. Organizations are interesting as they resemble very closely autopoietic systems. Also, organizations are at the center of many theorems in COT. This theory and RAF sets deal with closure. While one makes no distinction between catalysts and metabolites, the other one lacks the notion of time, which are essential elements of living systems. In the next paragraph we will show an relation between these two theories.

## Kinetics in RAF sets

If a theory is to have impact on real biochemical world, it must deal with the notions of that domain, thus, to gain a full understanding of closure we must complement the purely algebraic nature of RAF sets with ideas taken from Metabolic Control Analysis (MCA), a field generated to understand and measure fluxes in biochemical systems which is of common use in the field of metabolic engineering.

Current MCA is a quantitative theory that does not consider closure, as catalysts (i.e. enzymes) are placed in the network, but the reactions generating them are not taken into account. By putting the quantitative aspects of MCA and applying them to RAF sets, side by side, we can gain insight in how to study closure quantitatively. All the theories of metabolic closure (Autopoiesis,  $(M, R)$  systems, Autocatalytic sets, etc) are essentially algebraic or conceptual

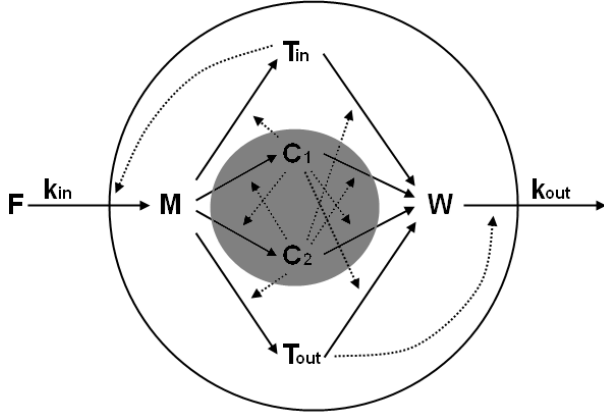


Figure 1: A simple example of a RAF set. Food elements  $F$  are incorporated into the system and generate metabolites  $M$ , which are transformed into two different catalysts sets: a)  $T_{in}$  which regulates the inflow of  $F$  and  $C_2$  which catalyzes its generation and b)  $T_{out}$  which regulates the outflow of waste metabolites  $W$  and  $C_1$  who catalyzes its formation. In addition,  $C_1$  and  $C_2$  catalyze the formation and destruction of the transporter catalysts ( $T_{out}$  and  $T_{in}$ , respectively), and also they mediate the generation and consumption of each other, forming the Reflexive Autocatalytic core of the system. Finally, growth is regulated by modulating the inflow of  $F$  and the outflow of  $W$ . We want to highlight the loop defined by metabolites  $M$  which turn into  $C_2$  who regulates the formation of  $C_1$  starting from  $M$ , a reaction regulated by  $C_2$ .

models centered on connectivity but not in dynamics. To go further in our understanding we must include the time course evolution of the concentrations inside the system.

Fortunately, the formalism of Reder (1988) that uses the stoichiometric matrix and the matrix  $D_x v$  to study rates, can be applied almost *verbatim* to analyze if a RAF set will grow or disappear. The great advantage of applying MCA formalisms is that we can quantitatively study how a system with metabolic closure can grow or disappear.

RAF sets are sets of coupled biochemical reactions with the attribute that the catalytic dependences between reactions and their catalysts are explicitly given. As said, RAF sets demand that almost all the molecules that conform a system can eventually be generated, directly or indirectly, from certain food materials and that all catalysts are produced by the system.

The *transformation* part of a RAF set can be represented by the formalism of the stoichiometric matrix, a well known tool extensively used in fields like MCA and Systems Biology in which every reaction is written as a column and every metabolite is referred to as a row. For example, the matrix  $N$  of the system described in figure 1 would be as following:

$$N = \begin{pmatrix} 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & -1 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

By using the column representation of reactions, it is convenient to define the addition of reactions as a standard addition operation of vectors. This operation expresses the occurrence of both reactions as a single net reaction.

Note that the *catalytic* part lies outside the stoichiometric matrix and cannot be deduced from it. But, in an idea that can be traced back at least to Reder (1988), the *catalytic* part can be represented by a matrix  $D_x v$  (also known as the Jacobian of the system) that contains all partial derivatives relating every reaction with every metabolite (or catalyst) in the system. Thus, the catalysts for a given reaction can be discovered by ranking the partial derivatives of the rate of this reaction respective to all metabolites (molecules) in the system. For example, the Jacobian matrix  $D_x v$  of the system described in figure 1 would be:

$$D_x v = \begin{pmatrix} \partial_M v_1 & 0 & 0 & 0 & \partial_{T_{in}} v_1 & 0 \\ \partial_M v_2 & 0 & \partial_{C_1} v_2 & \partial_{C_2} v_2 & 0 & 0 \\ \partial_M v_3 & 0 & \partial_{C_1} v_3 & \partial_{C_2} v_3 & 0 & 0 \\ \partial_M v_4 & 0 & \partial_{C_1} v_4 & 0 & \partial_{T_{in}} v_4 & 0 \\ \partial_M v_5 & 0 & 0 & \partial_{C_2} v_5 & 0 & \partial_{T_{out}} v_5 \\ 0 & \partial_W v_6 & \partial_{C_1} v_6 & \partial_{C_2} v_6 & 0 & 0 \\ 0 & \partial_W v_7 & \partial_{C_1} v_7 & \partial_{C_2} v_7 & 0 & 0 \\ 0 & \partial_W v_8 & 0 & \partial_{C_2} v_8 & \partial_{T_{in}} v_8 & 0 \\ 0 & \partial_W v_9 & 0 & \partial_{C_2} v_9 & 0 & 0 \end{pmatrix}$$



Every RAF set can then be described by two matrices;  $N$ , that shows the network connectivity, and  $D_x v$ , that quantifies catalyzations.

The necessity of using the  $D_x v$  matrix to analyze RAF sets lies in the fact that autocatalysis is a phenomenon that does not depend *only* on connectivity. As it has been shown recently by Plasson et al. (2010) and Piedrafito et al. (2010), the stability of an autocatalytic set depends on the relative rate of some reactions. Thus, two systems with identical connectivities but with different kinetics for some reactions can have vastly different behaviors.

As stated above, another theory concerned with formalizing biological organization is COT, a theoretical framework also centered in the idea of closure differing from RAF as the idea of catalyzation, perhaps the hallmark of RAF sets, is not considered. On the other hand COT brings an idea, the importance of fluxes in a network, that are not considered in RAF sets which is a purely algebraic approach to the description of biological organization. Thus an important question arises: can these two models be related? Can they support each other, in the sense of across fields fertilization? In the next section we clarify some relations between these two models.

An observation needed at the very beginning is that analysis using RAF sets and COT belong to two very different viewpoints as crucial elements in one theory are totally absent in the other. Thus as organizations (in the sense of COT) require that the overall flux across a relevant subset of reactions is maintained (thus avoiding the disappearance of crucial metabolites that, if absent, will produce network collapse). A mirror like situation can be stated with respect to catalyzations, a cornerstone idea in RAF sets, and (surprisingly) an idea that is absent from COT. Thus we should expect that if a system is a RAF set it is not immediate that it is also an organization. Only in some especial conditions we should be able to find how these ideas can be concurrently applied.

## A hidden relation between RAF sets and COT

A further dissection of RAF sets shows that, although fluxes and reaction rates initially seem to be absent from this model, kinetic ideas do exist just below the surface. In effect we propose two lemmas and a theorem that will bring new light to the problem of comparing both approaches:

**Lemma 1** *If a catalytic reaction system  $\mathcal{L} = (X, \mathcal{R}, C)$  is  $F$ -generated, then for all metabolites  $x$  (including catalysts) produced by any reaction  $r \in \mathcal{R}$ ,  $x \in \text{supp}(\mathcal{R})$  there is a positive linear combination of reactions  $\bar{r}_x = \sum_i \alpha_i r_i$  such that the metabolite  $x$  belongs to the products of the reaction  $\bar{r}_x$ ,  $x \in \pi(r_0)$  and the reactants of  $\bar{r}_x$  belong to the Food set, i. e.,  $\rho(r_0) \subseteq F$ .*

**Proof:** Considering the algorithm used to find the closure of  $\mathcal{L}$  (Hordijk and Steel, 2004), let  $W = F$ . Then add

the products of reactions  $\mathcal{R}_0 = \{r \in \mathcal{R} | \rho(r) \subseteq W\}$  to  $W$ . Adding all reactions in  $\mathcal{R}_0$  gives a global net reaction  $\bar{r}_0$  that consumes metabolites from the Food set  $F$  only and produces each metabolite in  $W$ . If this process is repeated, considering  $W = F \cup \pi(\mathcal{R}_0)$ , it is possible to build the set  $\mathcal{R}_1$  that contains all reactions that have their reactants in  $W$ , but excluding the reactions from  $\mathcal{R}_0$ . Adding all reactions in  $\mathcal{R}_1$  we obtain a new reaction  $\bar{r}'_1$  that requires metabolites from  $W$  only and produces any metabolite in  $\pi(\mathcal{R}_1)$ . To obtain the fact that this last reaction uses only metabolites from the Food source, let  $\bar{r}_1 = \alpha \bar{r}_0 + \bar{r}'_1$ , where  $\alpha$  is the most negative stoichiometric coefficient of the reaction  $\bar{r}'_1$ . This procedure takes enough metabolites from  $F$  to generate  $\pi(\mathcal{R}_1)$ . If we repeat this algorithm until it is not possible to find new metabolites, we will have generated  $cl_{\mathcal{R}}(F) = W$ . If the system is  $F$ -generated, according to Hordijk and Steel (2004), we have that  $cl_{\mathcal{R}}(F) = F \cup \text{supp}(\mathcal{R})$ . We have shown that for every metabolite  $x \in cl_{\mathcal{R}}(F)$  a composite reaction exists which generates it consuming food items only, in fact it is one of the  $r_i$ .

**Lemma 2** *If a catalytic reaction system  $\mathcal{L} = (X, \mathcal{R}, C)$  is  $F$ -generated, there is a strict positive linear combination of reactions  $\hat{r} = \sum_i \alpha_i r_i$  with  $\alpha_i > 0$  such that all metabolites are products of this reaction, i. e.,  $\hat{r}$  is a strictly positive vector.*

**Proof:** From Lemma 1 it follows that for each metabolite  $m_j$  there is a positive linear combination of reactions  $\bar{r}_{m_j} = \sum_i \alpha_{i,j} r_i$  such that this metabolite is produced exclusively from the Food set. This linear combination  $\bar{r}_{m_j}$  is the resultant net reaction associated with the path of reactions  $j$  that generate each metabolite  $m_j$ . If for all metabolites we add their generating reactions  $\bar{r}_{sum} = \sum_i \alpha_i r_i$  with  $\alpha_i = \sum_j \alpha_{i,j}$ , we have from the Lemma 1 that  $\bar{r}_{sum}$  is strictly positive. We must note that not all reactions will be used. We refer to these reactions generally as  $r_s$ , having  $\alpha_s = 0$  in  $r_{sum}$  for those reactions. If we consider the sum of these reactions  $r_{not} = \sum_s r_s$ , it will consume metabolites. To maintain  $\hat{r}$  positive and still fire these non-essential reaction (this will be needed later), we set  $\hat{r} = \beta r_{sum} + r_{not}$  with  $\beta$  sufficiently large. Then we can construct a strictly positive linear combination  $\hat{r} = \beta r_{sum} + r_{not} = \sum_i \bar{\alpha}_i r_i$  with:

$$\bar{\alpha}_i = \begin{cases} 1 & \text{if } r_i \in \{r_s\} \\ \beta \alpha_i & \text{if } r_i \notin \{r_s\} \end{cases} \quad \text{note } \bar{\alpha}_i > 0$$

and all the metabolites are product of this reaction  $r_0$ .

These lemmas, framed completely in the language of RAF sets, could be interpreted as mere technical results about RAF sets. In essence they state that every metabolite can be generated from the food set and makes explicit the overall reaction producing each, non-food, metabolite. But every time we use a stoichiometric matrix  $N$  we are implying a given kinetics because of the necessary equation

relating  $N$  to the change of concentrations:  $N \cdot v = dX/dt$ , where  $v$  is the vector of rates. Thus the requirement, in COT, that  $(dX/dt \geq 0)$  can be phrased as a condition on the components of  $v$ . These lemmas show how some (not all) Organizations could be RAF sets, and it is interestingly that they are proved by using notions of linear algebra. Also note that the positive linear combination predicted by the lemmas explicitly shows how to combine individual reactions in any RAF set to attain mass-maintenance.

Once we have established this link we can a little bit further and search for deeper connections. The next theorem continues to exploit matrix  $N$  to sketch how some RA sets are F-generated using the stoichiometric matrix  $N$ .

**Theorem 1** *If a catalytic reaction system is F-generated, then there is a strictly positive rate vector  $v$ , such that  $N \cdot v = dX/dt$  is also strictly positive, where  $N$  is the stoichiometric matrix of the system.*

**Proof:** We note that the operation  $N \cdot v = dX/dt$  is equivalently mathematical to make a linear combination of reactions  $r = \sum_j \alpha_j r_j$  if we consider each reaction as a column and  $\alpha_j$  as the velocity of reaction  $r_j$ . From Lemma 2, if we take the reaction  $r_0$  and choose  $v_j = \bar{\alpha}_j$  (normalizing time units), therefore a  $v$  exists with components  $v_j > 0$  associated to a flux vector  $dX/dt$  that satisfies  $dX_i/dt > 0$ , equivalent to the column representation of  $r_0$  with all of their components also positive.

**Corollary 1** *If a catalytic reaction system is F-generated, then it is also an organization.*

**Proof:** An F-generated system is, by definition, *closed* and as theorem 2 shows, it also satisfies the property of *mass-maintenance*. Thus, it is an organization.

This theorem explains the existing relations between Organizations, F-generated sets, RA sets and RAF (see figure 2). Essentially, we have proved that all F-generated sets are *organizations* and a subset of them are also *reflexive autocatalytic*. This subset is the RAF sets. Theorem 1 is a simple one that has the virtue of illuminating how these two theoretical frameworks are related to each other.

This result is important because some new technical theorems have been obtained by Dittrich's group, for example, on how to detect organizations among real metabolic networks (Centler et al., 2010, 2008). Thus, our theorem shows that these new tools, developed to find organizations, could be also used to search for RAF sets.

In addition, we will make a definition to the sets that are organizations and RA at the same time.

**Definition 1** *If a catalytic reaction system is Reflexive Autocatalytic and an Organization, then it is a Reflexive Autocatalytic Organization, RAO.*

These sets are reaction systems that can be sustained, but not necessarily can be generated from a food set  $F$  exclusively. We have shown that all F-generated sets are orga-

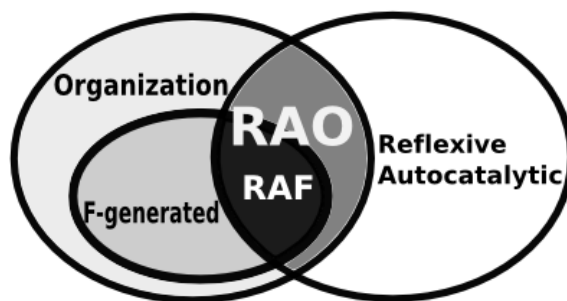


Figure 2: Venn's diagram depicting the logical relations between RA, RAF, and F-generated sets and *organizations* under COT's definition. All RAF sets are organizations, but whether all organizations are F-generated is an ambiguous matter.

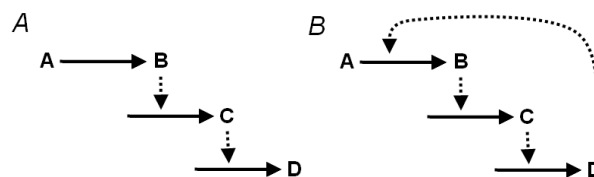


Figure 3: A: A chain of dependent catalyzations. B: A catalyzation loop

nizations, but the converse result (all organizations are F-generated sets) is more difficult to handle. We propose two different approaches: First, if one decides that the Food set corresponds only to the molecules generated from the empty set (in COTs phrasing of reactions), then it is clear that there are organizations which are not F-generated. On the other hand, for any organization it is always possible (due to the closure property) to find a suitable set (generally not unique)  $F$  such that the corresponding F-generated set is equal to the given organization. Thus, the extend to which organizations and F-generated sets overlap depends on which approach one takes to express COT systems in terms of RAF sets.

## The Loop Theorem

As most theories on biological organization are centered in the notion of closure (Letelier et al., 2011), RAF sets formalism give a succinct and useful description of closure. First, we shall consider a *chain* of catalyzations in which a product from one reaction catalyzes another reaction in the chain (figure 3A). If eventually a product catalyzes an earlier step (figure 3B), we have a catalyzation loop. As we shall see, in a RAO it is always possible to find such a catalyzation loop if the catalysts are not part of the food set. This condition seems natural for systems with metabolic closure.

Considering this definition we propose:

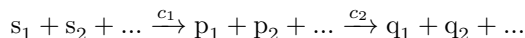
**Theorem 2** *If all catalysts in a RAO are generated by the system, then there is at least one catalyzation loop.*

**Proof:** In such a RAO every catalyst must be generated by a reaction, which in turn must be catalyzed too. In this sense, the production of every catalyst is directly dependent on another and indirectly dependent on a sequence of catalysts. The number of catalysts is finite therefore, at some point a catalyst must depend indirectly on itself, thus, forming at least one catalyzation loop. Note that not every catalyst is part of a loop as it is allowed that some catalysts may catalyze reactions which yield no catalysts as products, yet the system as a whole must have at least one catalyzation loop. Note that in case of direct autocatalysis, the loop is trivial. Also, whether there is more than one catalyzation loop is a question that must be addressed in each particular case.

An unsuspected consequence of the loop theorem is that some of the catalysts inside the catalyzation loop must have a dual catalytic role, that is enzymes that catalyze at least the creation of other two enzymes, if not happen the trivial case of all enzymes catalyze the creation of another one enzyme. This is interesting, as one modern re-interpretation of Rosen's results about how living systems avoid infinite regress is by having enzymes with dual functions (Letelier et al., 2006). Thus, this systemic result (i.e. existence of moonlighting enzymes) can be achieved by two different methods.

This theorem is a basic result that follows directly from the basic definitions of RAOs, but it shows an important property that needs to be underlined: the catalyzation loops (one or more) inside a RAO may be considered as its autocatalytic core and, functionally, there is a difference between the catalysts of the loop and the ones outside it.

We conjecture that the functional segregation hinted has important consequences. In effect, to confer stability to the core the catalysts outside it control the inflow and outflow of matter to and from the core. Thus, the net flow of matter inside it must be controlled, as a large flow would generate an exponential runaway and a small one would extinguish some core components, destroying its organization. Keeping this balance between in and outflow will be seen as homeostatic regulation. In summary, we conjecture that the catalyzation loop confers long term stability to the network. The analytical proof of this result seems difficult, but we did computer simulations in small (toy-like) systems and using mass-action kinetics, expresed for reactions:



By the formula:

$$\frac{d[p_k]}{dt} = k_1[c_1] \prod_i [s_i] - k_2[c_2] \prod_j [p_j]$$

Figure 4 shows one example for the temporal evolution of the concentrations of molecules for the RAF system of

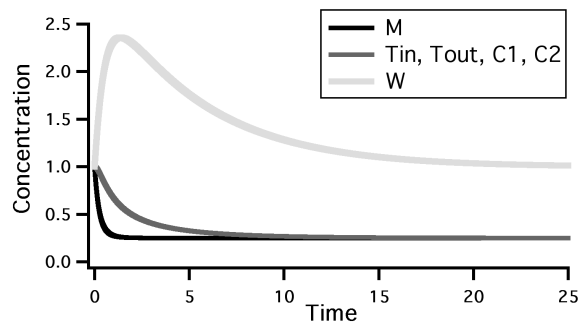


Figure 4: Temporal evolution of RAF toy system (see figure 1). The system reaches a steady state in which all concentrations are different from 0.

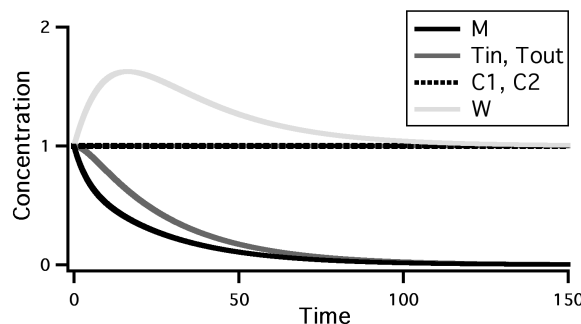


Figure 5: Temporal evolution of a non-RAF system. Although the concentrations of the catalysts  $C_1$  and  $C_2$  were fixed, the system decays until its components disappear.

figure 1. We can see that a steady state is achieved. In figure 5 we simulated a similar system but without the reactions that generate or destroy the catalysts  $C_1$  or  $C_2$ , removing the catalyzation loop and making the system a non-RAF set. In this last case the concentrations of many components decay to zero, stopping the network dynamics.

We also did the bifurcation analysis by varying rate constants  $k_{in}$  and  $k_{out}$  corresponding to the reactions  $r_1 : \phi \rightarrow M$  and  $r_9 : W \rightarrow \phi$  respectively. For the RAF set, almost every combination of parameters  $k_{in}$  and  $k_{out}$  leads to a steady state, except at the border where  $k_{in} = 0$  or  $k_{out} = 0$  and some regions close to these.

On the other hand, the non-RAF set has no stable points for most values of  $k_{in}$  and  $k_{out}$ . This puts in evidence the relevance of the autocatalytic core, so that the growth of one part of the system encompasses the rest and grows harmonically and coherently between the inflow and outflow of matter.

## Growth and Homeostasis in Autopoietic Systems

Any increase in the concentration of a loop catalyst will translate into an increased concentration of every other cata-

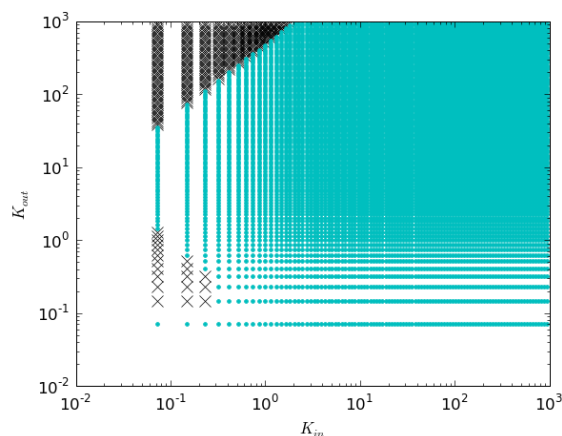


Figure 6: Bifurcation diagram of the RAF set in figure 1, for variables  $k_{in}$  and  $k_{out}$ . A similar diagram for a non-RAF system has no stable region. (cyan dots=stable, black Xs=unstable).

lyst in the loop, which would consequently lead to a further increment of the first one, exhibiting an apparent autocatalytic behavior. At the same time, this loop must be connected to side branches that lead to the production of catalysts that are not directly related to autocatalysis, but with the obtention and processing of the food sources that sustain it. An interesting type of branches are the ones that lead to the regulation of the enzymes that control the in and outfluxes, because they are supposed to regulate the whole metabolisms growth rate by coordinating these fluxes. This fact shows us the importance of the topology, because every enzyme must be whether part of a loop or a branch of it, thus, a change in an enzyme concentration which is part of the loop will have repercussions in the whole systems growth, as it also affects the enzymes that regulate the fluxes. Thus, the RAF sets may help to understand the dynamics of the homeostatic process. This is not mutually exclusive with the fact that an increase in the concentration of an enzyme outside the loop may have a direct repercussion on global growth.

The loop theorem has an important application for autopoietic systems, that can be defined as self-encapsulated RAF sets. For an autopoietic system (that according to the above theorem must contain at least one autocatalytic loop) to be stable in time, there must be a fine balance between the generation and destruction of molecules. But there must also be a balance respective to control its volume in order to keep the concentrations unaltered. Thus, the organization (*à la* Autopoiesis) of a RAF set must be under a precise homeostatic control, as growth must be promoted, but in the context of compensating for the volume increase without suffering the consequences of autocatalytic growth. Thus, in a first

approach, we must allow for a system to grow in terms of the net amount of molecules, but not in concentration. This implies that volume must be under active control and that allowing the system to grow would not be a contradiction with homeostatic principles.

## Discussion and Conclusion

As we have previously stated (Jaramillo et al., 2010) we conclude again that RAF sets formalism is particularly suited to study closure. Of course many aspects of metabolic closure escape this theory (the operator of organizational invariance  $\beta$  of  $(M,R)$  systems is a prime example), but this framework provides a solid starting point. The loop theorem proved here, which is a property shared by RAOs, Autopoietic and  $(M,R)$  systems is a good example of its power.

Another important point of the present study is to apply the analysis of COT to RAF sets. As it is usual in theoretical biology, the different frameworks generated to explain living organization exist in closed universes without dialogue between competing theories. Here, we partially break this isolation by showing how organizations *à la* COT contain all RAF sets, but not all RA sets. This inclusion, although obvious and expected from a theoretical viewpoint, is not easy to prove. We have developed demonstrations using arguments from linear algebra, instead of the set theory arguments favored in RAF. The most unexpected result is the uncovering of chemical kinetics arguments in RAF sets. In effect, RAF sets appear to be a purely algebraic entity, without considerations for time nor kinetics; but as soon as their stoichiometric matrix is expressed, the kinetic arguments of COT are made obvious. Thus our lemmas and theorems show deep relations between the pure algebraic formulation of RAF sets with the dynamics of organizations in COT. Perhaps this same reasoning could be also be applied to  $(M,R)$  systems. Taken together, the results shown here show the value of putting all the different notions of metabolic closure under a common analytical umbrella.

COT has already produced an interesting number of results on the dynamics of reaction networks, in particular regarding to the long-term temporal behaviour and stability of these systems (Dittrich and Di Fenizio, 2007). An interesting result from this theory, which complements the loop theorem presented here, is the decomposition theorem for organizations (Veloz et al., 2011). This theorem states that, under certain conditions, it is possible to split a system into subsystems whose dynamic behaviour are weakly coupled. Thus, an open question is to investigate how our loop theorem, which seems to indicate that systems cannot be segmented, is compatible with such uncoupling of subsystems. In effect a catalyzation loop might constitute the minimum decomposable unit.

We presented the conjecture that systems with at least one catalyzation loop are more stable than similar systems without such loops. This is a powerful result that will unavoid-

ably demand tools from MCA, the most elaborate theory about fluxes in biochemical networks, to be proven or refuted.

In summary, our efforts show that closure is a conceptual key to understand biological organization, as an example we have come close to use closure as an argument to prove one theorem (loop theorem), which we believe is a valuable conceptual step and a fertile direction for theoretical biology.

## Acknowledgements

We would like to thank the reviewers for their comments. They were very helpful to improve this manuscript. Funded by FONDECYT grant 1110247, CONICYT scholarship 22101236 and 22110804.

## In memoriam

This work reflects the school of thinking generated by Francisco Varela in Chile during the 70's. Francisco was a valued friend, a guide and an inspiration for new generations of biologists.

## References

- Centler, F., Kaleta, C., Speroni di Fenizio, P., and Dittrich, P. (2008). Computing chemical organizations in biological networks. *Bioinformatics*, 24(14):1611 – 1618.
- Centler, F., Kaleta, C., Speroni di Fenizio, P., and Dittrich, P. (2010). A parallel algorithm to compute chemical organizations in biological networks. *Bioinformatics*, 26(14):1788 – 11789.
- Dittrich, P. and Di Fenizio, P. S. (2007). Chemical organization theory. *Bulletin of Mathematical Biology*, 69:1199–1231.
- Hordijk, W. and Steel, M. (2004). Detecting autocatalytic, self-sustaining sets in chemical reaction systems. *Journal of Theoretical Biology*, 227:451–461.
- Jaramillo, S., Honorato-Zimmer, R., Pereira, U., Contreras, D., Reynaert, B., Hernández, V., Soto-Andrade, J., Cárdenas, M., Cornish-Bowden, A., and Letelier, J. (2010). (M,R) Systems and RAF sets: Common ideas, tools and projections. *Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*, pages 94 – 100.
- Kauffman, S. (1993). *The Origins of Order*. Oxford University Press, New York.
- Letelier, J. C., Cárdenas, M. L., and Cornish-Bowden, A. (2011). From *L'Homme Machine* to metabolic closure: steps towards understanding life. *Journal of Theoretical Biology (in print)*.
- Letelier, J. C., Soto-Andrade, J., Guíñez Abarzúa, F., Cornish-Bowden, A., and Cárdenas, M. L. (2006). Organizational invariance and metabolic closure: analysis in terms of (M,R) systems. *Journal of Theoretical Biology*, 238:949–961.
- Maturana, H. and Varela, F. (1980). *Autopoiesis and Cognition: the Realisation of the Living*. D. Reidel Publishing Company, Dordrecht, The Netherlands.
- Piedrafitra, G., Montero, F., Morán, F., Cárdenas, M. L., and Cornish-Bowden, A. (2010). A simple self-maintaining metabolic system: Robustness, autocatalysis, bistability. *PLoS Comput Biol*, 6(8):e1000872.
- Plasson, R., Brandenburg, A., Jullien, L., and Bersini, H. (2010). Autocatalyses. *Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*, pages 4 – 11.
- Reder, C. (1988). Metabolic control theory: A structural approach. *Journal of Theoretical Biology*, 135(2):175 – 201.
- Rosen, R. (1958). The representation of biological systems from the standpoint of the theory of categories. *Bulletin of Mathematical Biology*, 20(4):317–341.
- Veloz, T., Reynaert, B., Rojas, D., and Dittrich, P. (2011). A decomposition theorem in chemical organizations. *Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems (this volume)*.

# From egocentric systems to systems allowing for Theory of Mind and mutualism

Holk Cruse<sup>1</sup> and Malte Schilling<sup>1,2</sup>

<sup>1</sup>Biological Cybernetics and Theoretical Biology Department,  
and Center for Cognitive Interaction Technology (CITEC), University of Bielefeld, Germany

<sup>2</sup>ICSI Berkeley  
holk.cruse@uni-bielefeld.de

## Abstract

Simple artificial agents representing more or less elaborated Braitenberg vehicles, usually adopt an egocentric view. One example is Walknet, a biologically inspired neural network controlling hexapod walking. Here we show how such a controller can be expanded to be able to interpret observed behaviours that are performed by other individuals, i.e. the system shows properties of a mirror system. This allows to further expand the network to become an “allocentric” system that might implement subjective feelings which could be attributed to other individuals, i.e. the system implements a Theory of Mind. As a last expansion we introduce a two-body model, or we-model, which may allow for mutualism. Application of we-models allows for what often has been called the third person’s view. The different steps proposed can be interpreted as corresponding to an evolutionary development.

## Introduction

Artificial agents being based on natural creatures may usually be characterized as to hold an ‘egocentric’ view: in such agents, the sensory input is related to the own body representing the center of the agent’s world. Correspondingly, motor output activities are based on the own geometrical—and possibly mental—position. Here we attempt to introduce a way how the controller of such an autonomous agent may be changed to allow the agent to ‘put itself into the partner’s shoes’, in other words to allow for theory of mind (ToM), and to show empathy. A further goal is to develop a (neuronal) control structure that may form the basis of mutualism, i.e. the faculty to cooperate with a partner using shared goals (Tomasello, 2009). Such a control structure may serve as a quantitatively defined hypothesis and may as such help to understand the underlying mechanisms of the corresponding biological system.

When attempting to simulate higher mental functions as are specific memory systems, attention, cognition or consciousness, for example, authors do, in general, not apply a whole-systems approach, but instead consider specific networks suited to represent the specific function of interest. Therefore, in many cases, it remains open how these specific networks may be embedded into the complete system, i.e. how the different networks are switched on or off and how these local networks receive input from and provide output for the complete system. To avoid this problem, we take a whole-systems approach. We investigate such phenomena under the

condition that these networks are embedded into an autonomously behaving agent, i.e. an agent equipped with a body characterised by many parallel and serially arranged degrees of freedom and a control network containing a set of preexisting reactive behaviours.

Schilling and Cruse (submitted) have proposed a network that has been worked out in more detail and called *reaCog* (this work is based on the reactive control system Walknet (Dürr et al., 2004) and the cognitive extensions have been introduced in Cruse and Schilling (2010)). This network is able to control a hexapod system by applying a structure consisting of two levels. The lower level is endowed with properties that correspond to insect-type behaviours (as are walking, climbing and navigation), about which already detailed knowledge is available (Dürr et al. 2004, Bläsing 2006, Wehner 2008). This level is based on a reactive, or behaviour-based, architecture, i.e. a collection of local, in general recurrent, neural networks (RNN). The second level of *reaCog* concerns an expansion allowing for the introduction of cognitive abilities as explained below. Generally, the architecture of our system is not based on the idea to consist of one holistic RNN, but represents a localist approach the advantages of which are convincingly advocated by Cooper and Shallice (2006).

When starting with an insect-like body and insect-inspired behaviour-based networks we do not imply that insects were endowed with higher cognitive functions as are metacognition, ToM or consciousness, although already in insects a number of astonishing properties can be found which by some authors are called cognitive (e.g. application of concepts like symmetry, sameness or protocounting, see Menzel et al., 2007). However, we assume that any cognitive system is strongly relying on such reactive—or behaviour-based—structures. Different to a reactive system, a cognitive system in the strict sense should be able to exploit stored information independent of the context in which this information has been acquired. This means, a cognitive system should be able to combine existing memory elements in a new way and use these new combinations for controlling behaviour and planning ahead. As we have shown by having developed *reaCog*, only a limited number of expansions are required to reach such a cognitive level (Cruse and Schilling 2010; Schilling and Cruse, submitted). The most important expansion concerns the introduction of a ‘manipulable’ body model. In order to be able to plan ahead, this internal model of

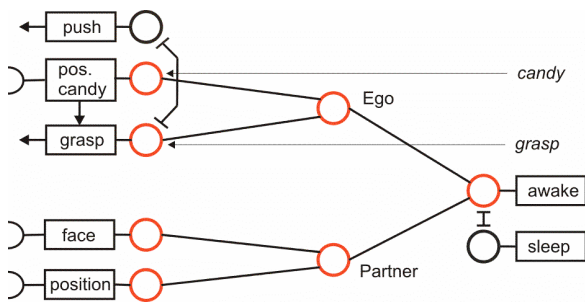


Fig. 1. An egocentric network represents the situation “Ego grasp candy”. The figure shows a section of the network reaCog (Schilling and Cruse, *subm.*). Local networks are symbolized by rectangles and names. Motivation units are shown by circles (connection to the corresponding network see Fig. 4). Active motivation units are marked by red colour. Arrows represent excitatory connections, T-shaped connections are inhibitory. Visual and proprioceptive input is marked by the half-circles, left side. Acoustic input representing words is shown by italic letters at the right side.

the own body (plus some aspects of the world, e.g. an obstacle) is required to internally simulate different behaviours in order to test whether this specific behaviour is suited to cope with an actual problem. The second expansion concerns an attention system. This system consists of two layers, a spreading activation layer (SAL) and a winner-take-all layer (WTA). This two-layer network enables the agent to select a specific behavioural element, which is normally not activated in the actual context. Via internal simulation, the system can then test whether this newly selected behavioural element is suited to solve the problem at hand, a procedure that has been termed “probehandeln” following Freud (1911). New behaviours found by this procedure and that, by means of the simulation and the subsequent behavioural test, prove to be adaptive will be stored in the long-term memory, thereby enriching the behaviour-based architecture. As for a well designed reactive system new problems may occur only rarely, reaCog can be regarded a reactive system that exploits its cognitive properties only for short periods of time required to solve a problem at hand.

Based on the ideas of Narayanan (Narayanan, 1997 and Feldman and Narayanan, 2004) and Steels (1995, 2003) we have further designed a simple expansion of reaCog that allows connecting behavioural elements of this system with so called word nets (RNNs representing an individual verbal expression, e.g. “leg”, or “swing”) that carry the corresponding meaning (Cruse, 2010). Therefore, the symbols are grounded (Steels, 2003) allowing the agent to ‘understand’ the meaning of such a word when given to the agent.

Like most other autonomous systems, reaCog holds an “egocentric” view. The agent might be able to recognize and represent objects. We further assume that the agent can also recognize, as a specific kind of object, a conspecific (see Steels and Spranger, 2008 and Spranger et al., 2009 for solutions). In addition we assume that the agent can attribute properties to the object or the partner (e.g. a face, a spatial position). All these expansions, however, do not enable the agent to “put himself into the partner’s shoes”. In other words,

the agent is not able to realize that the partner may see the agent himself as having a property (e.g. a position). Thus, in this network there is no possibility to represent the change of roles (“If I were him”). In other words, the capability to have a ToM is lacking. A classical procedure for testing whether an agent allows for the ability of ToM is the so called Sally-Anne task. Two subjects are shown that a candy lying on the table is hidden under a black cover. Then one subject, Sally, has to leave the room whilst the candy is now hidden under the white cover, as observed by Anne. After Sally has come back, Anne is asked under which cover Sally will probably search for the candy. If Anne points to the black cover, she is assumed to have ToM, but not, if she points to the white cover where the candy really is placed.

The network reaCog even less shows the ability to perform mutualistic behaviour (Tomasello, 2009), i.e. to develop shared goals and to try to follow them, even when the individual agent may receive no specific advantage. A simple example is when two individuals are trying to carry a load, for instance a table through an environment containing obstacles. In the reminder we show how reaCog can be expanded to endow the agent with these capabilities. To be in a position to explain the structures and their properties in an easily understandable way, we illustrate the expansions of reaCog by attempting to maintain the number of neuronal units as small as possible. In this way we hope to provide a functional understanding of how systems able to develop a ToM and later a structure allowing for mutualism may have arisen from an egocentric system. The different steps introduced might represent a hypothetical evolutionary sequence.

## The Model

To simplify the description, we will focus on a small section of reaCog as illustrated in Fig. 1. Basically, the network consists of sensorimotor networks, or memory elements, connected with motivation units. In the figures, the networks are indicated by rectangles with verbal descriptors. Motivation units (depicted as circles) can adopt an activation value within the interval  $[0,1]$ . In the figures, activated units are marked as red circles, inactive ones are shown as black circles. Two of these motivation units may either be connected via (mutual) inhibition or via (mutual) excitation, or not be connected at all. Groups of excitatorily connected units stabilize each other. I.e. when one unit of such a group is activated, all the members of that group will become activated, too, except for those units that are connected via mutual inhibition. These inhibitory connections form a local winner-take-all (WTA) net with the consequence that only one of these units will stay active over some iterations. Two such motivation units represent the state Awake and the state Sleep, respectively. In the awake state, several sensory or motor elements are activated. These elements may form different contextual groups. Here we focus on two such groups. One group refers to external objects, in this case a conspecific (“partner”), represented by the memory elements “face” and “position”, which stand for the visual appearance and spatial location of the partner to be recognized by the corresponding networks. Together with the unit Partner these motivation units form an excitatory network. The elements of the second group refer to the agent. The agent can select between a number of actions



(in Fig. 1 “push” and “grasp”), the motivation units of which are connected via mutual inhibition (connections with T-shaped endings). The agent is also assumed to recognize an object, a candy lying on the table. Fig. 1 shows a memory element representing the position of the candy (*pos.candy*) relative to the agent. The agent may also be equipped with a network representing the experience of pain, which is connected to any specific body position, but this faculty will only be explained later. The motivation unit connecting the agent-related elements has been called Ego unit in the figures. To avoid a possible misunderstanding, it should be made clear that this name represents only a technical term and should not be understood as to mean that the agent has any kind of self-knowledge. As mentioned, the system may also be equipped with word nets that allow to recognize verbal statements as “*grasp*” or “*candy*” or “*partner*” which, if stimulated, activate the corresponding sensorimotor networks (in the figures these inputs are indicated by the terms given in *italic*; the word nets themselves are not shown).

Of course, any partner, if being equipped with a corresponding network, may likewise recognize our agent, but, as mentioned, the agent does not know this.

The behaviour-based—or sensorimotor—RNNs indicated by rectangles in the figures might be realized as simple associators connecting a sensory input with a motor output (Dürr et al., 2004; Cruse and Wehner, 2011) and may function as an implicit body model, that can be used to control the behaviour by computing the inverse kinematics. Alternatively, as conceptualized in *reaCog* (Cruse and Schilling, 2010; Schilling, 2011; Schilling and Cruse, *subm.*), sensorimotor RNNs may be connected to an explicit body model. In this case, the network is equipped with a switch that allows to turn on or off the motor output to either control the behaviour or instead to activate only the body model and in this way simulate the behaviour. In the latter case, the system may be termed to imagine this behaviour.

To realize the motivation units and RNN units we use the so called Input Compensation (IC) units, type suppression units (Kühn et al. 2007, Makarov et al. 2008), first, because a simple learning algorithm is available to train such networks. Secondly, because such networks maintain the input activation as long as the input is provided, but, if trained to hold a static attractor, also after the input is switched off. A motivation unit that is connected to a behaviour-based RNN, controls the output of its network by multiplying the output by its activation value (see below, Fig. 5). In this way, a motivation unit when activated may be called to ‘open’ the corresponding network (representing a top-down influence). As will be mentioned below, sensorimotor networks may also be used to respond to sensory input. In this case, the network showing the best fit to the actual sensory input (or the smallest error) will activate its motivation unit (this bottom-up influence is not depicted in Fig. 5). In the simulation proposed here, only the motivation unit network has been studied (for an explicit simulation of such a network see Cruse and Wehner, 2011).

*Phenomenal aspect:* Before we continue to describe the property of the network in more detail, a fundamental, and unsolved problem has to be addressed. When trying to understand a cognitive system the question arises how a neuronal system representing a physical structure is able to

allow for the faculty to experience subjective feelings, an example is feeling pain. This subjective or phenomenal aspect is relevant for (at least some) living systems. What is the problem? We can easily think of neuronal structures that, activated by nociceptors, for example, may produce chemical substances or activate specific behaviours (e.g. withdrawal or speech acts), i.e. form a series of causally connected physical states. But there is no concrete idea how (and why) the fact that these (or some of these) physical activities are accompanied by the feeling of pain, i.e. the subjective aspect, may be reified. The problem of understanding the relation between the physical aspect and the phenomenal aspect has eventually been termed the ‘hard problem’ (Chalmers, 1996) and will not attempted to be solved here. In order to be nevertheless able to use terms describing (or at least associated with) subjective feelings when discussing the properties of our network, we make the following assumption. An RNN as used here can adopt attractor states that are reached when the network has been given enough time for relaxation. In mathematical terms the attractor state can be defined as the so-called harmony value of the net reaching a maximum value. Following Cruse (1999, 2003) we assume that the activation of such a network is accompanied by subjective experience (or a phenomenal aspect) if the harmony value of the net has reached a given threshold, in other words, if the net has sufficiently well approached its attractor state. This hypothesis does of course not represent a solution of the hard problem, but nonetheless provides a way to operationalise the problem. Its function in this context is to allow us using terms associated with subjective or phenomenal aspects when describing states of our physical network. Using this hypothesis we are in a position to bridge the ‘explanatory gap’ on a descriptive level. If other mechanisms underlying the phenomenal aspect were found, they could replace our hypothesis without, as we believe, influencing the rest of the arguments.

## The functioning of the network – an example

The agent equipped with *reaCog*, the, for our discussion, relevant part of which is depicted in Fig. 1, is able to show the following simple behaviour. If we assume that elements “*grasp*” and “*pos.candy*” are activated by an external verbal command as indicated by thin arrows (in the figures marked by *italic* letters, e.g. Fig. 1 *grasp*, *candy*), this input will activate the motivation units *grasp* and *pos.candy*. The former will open the behaviour represented in the RNN *grasp* and activate the unit Ego. Further, the unit *pos.candy* when activated will open the RNN allowing to recognize the spatial position of the candy. The *grasp* network receives input from the *pos.candy* network that provides the information to the *grasp* network concerning the goal for the movement to be performed. Therefore, the movement can now be executed. As an alternative to verbally given input, the agent, after having registered the candy, may decide to perform a grasp movement, the decision being determined by its internal state requiring a network not shown in the figures. In the following examples we will however use verbal input only, because this simplifies explanation of the concepts proposed. As illustrated in Fig. 1, at the same time the agent may be able to represent a partner, characterized by its face and its position.

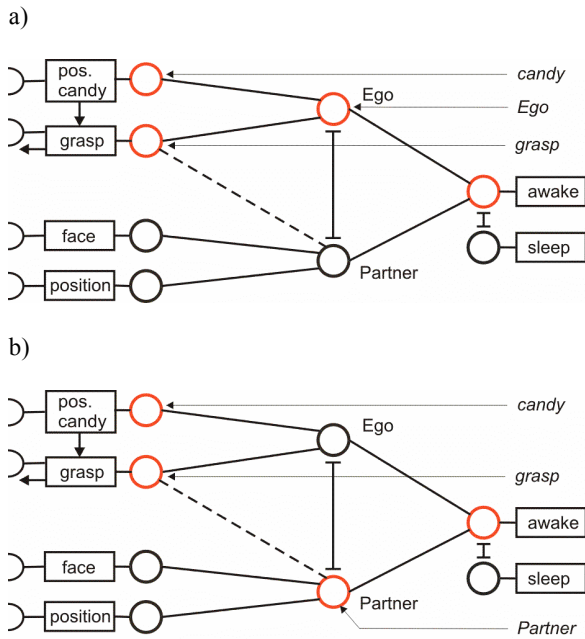


Fig. 2. An egocentric network representing the situation “Ego grasp candy” (a) and the situation “Partner is seen as grasping a candy” (b). The sensorimotor element “grasp” provides motor output and receives sensory (e.g. visual) input. Its units show properties corresponding to those of mirror neurons as it represents a circuit shared between the Ego and the partner units. See Fig. 1 for further explanation.

## Mirror systems

How may this network be changed to allow for ToM and mutualism? Several changes are proposed as will be illustrated in consecutive steps depicted in Figs. 2, 3 and 4.

A body model, apart from being used to control movement by calculating the inverse kinematics (Fig. 1), can also be used for a different purpose. When observing somebody else performing a grasp or a push movement, the visual input can be given to the body model which then can be used to simulate, or “internally copy”, the observed behaviour (e.g. “grasp”) following the “simulation theory” (e.g. Jeannerod, 2006 & 1999, Gallese & Lakoff, 2005). This application of the body model is suited to minimize errors when interpreting the (underspecified) visual input (e.g. Schilling, 2011). To symbolize this ability, in Fig. 2 the net ‘grasp’ is also equipped with sensory (visual) input. By application of a specific RNN forming a holistic system as has been proposed by Cruse and Schilling (2010) and Schilling (2011), one and the same body model is exploited for both purposes as are motor control and interpretation of sensory input. If a grasping movement is observed, the body model activates the element ‘grasp’. To allow the representation of the partner performing a grasping movement, too, we need another expansion, namely the introduction of connections between the unit representing the partner with (some of) the behavioural elements that, in the egocentric system (Fig. 1), are only

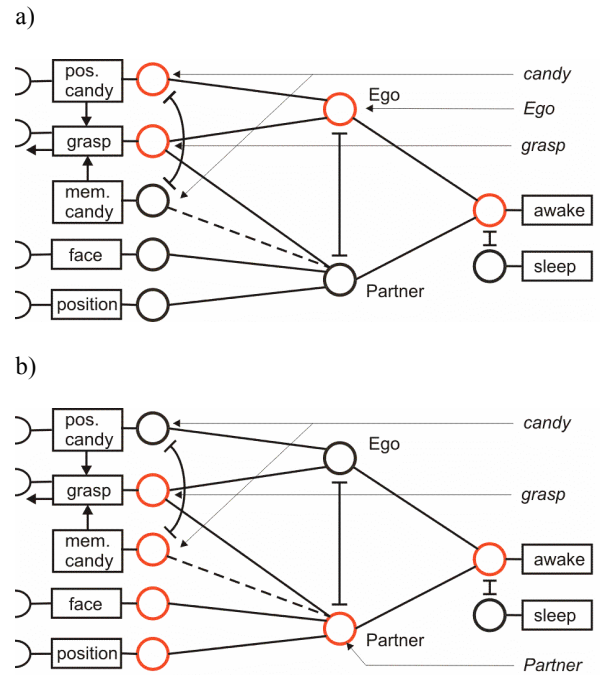


Fig. 3. A network being able to represent an egocentric view (a, situation “Ego grasp candy”) and the view as seen by the partner (b, situation “Partner grasp candy”), thus allowing for ToM. For further explanations see Fig. 1 and text.

connected with the Ego Unit. In our example this refers to element ‘grasp’ (see Fig. 2, dashed line). In addition, Unit Ego and unit Partner have to be connected via mutual inhibition (Fig. 2). This means that either unit Ego or unit Partner can be activated at a given moment in time.

With this network we can represent two situations: (i) if, as depicted in Fig. 2a, units Ego, grasp and pos.candy are coactivated, the network represents the agent to grasp the candy or to imagine such a grasping movement (the representation of this situation is already possible for the network shown in Fig. 1). (ii) However, the agent can also record a grasping movement of the partner. In this case, the sensorimotor element ‘grasp’ is activated together with the unit Partner, whereas unit Ego is inhibited. In Fig. 2b this situation is illustrated by motivation unit Partner shown in red and unit Ego in black. In both situations the neurons of the element grasp are activated. Such an architecture has eventually be termed to apply ‘shared circuits’ and strongly reminds of properties characterizing mirror neurons. Therefore, application of such shared circuits has been described as ‘mirroring’ (Keyers and Gazzola, 2011). Units of the grasp net represent to movement and its goal, and thus correspond to represent a motor act as attributed to mirror neurons (Rizzolatti and Luppino 2001). However, the goal in both cases (Fig. 2a, 2b) is represented as being viewed by the agent, not as being represented by the partner.

## Theory of Mind

Therefore, both circuits, as depicted in Fig. 1 and Fig. 2a,b still represent egocentric systems. We will now proceed allowing the agent to be able to simulate the behaviour and the internal view, including the sensory experience, of the partner, a property that has been characterized as ToM. To this end, we will present a simple simulation of the Sally-Anne task mentioned above. To be able to represent some aspects of the memory of the partner required for this task, in our network the unit **Partner** is given a connection to memory elements representing the position of the candy as viewed by the partner (Fig. 3, dashed line). Now imagine that subject Anne is either equipped with a network as depicted in Fig. 2 or in Fig. 3. Application of a system shown in Fig. 2 means that the agent (Anne) has only one representation of the candy's position, the one seen last. Therefore only this, correct, position can be activated and the partner is imagined to grasp the correct position as observed in children younger than about four years. The child is not taking into account the position the partner assumes. In contrast in a system as presented in Fig. 3a, there is a difference in thinking of oneself grasping the candy or the partner doing it. When the agent imagines itself to grasp the candy, it would grasp to the correct and known position. If asked to simulate the internal state of the partner, as is required in the case of the Sally-Anne test, (Fig. 3b), the position connected to the partner Sally will be used and the agent would rightfully deduct that the partners grasp would be directed towards this position which is wrong, but this fact is not known by the partner. Therefore, the network shown in Fig. 3 allows for ToM, in contrast to the network shown in Fig. 2. The critical difference between both networks is that the network shown in Fig. 3 contains a separate representation of (a part of) the partner's memory. Ishida et al. (2010) describe mirror neurons that are able to represent this property.

## Feeling pain

To illustrate another, more difficult case, let us come back to a push movement being directed to a partner. This case is more complex because roles can be interchanged in this scenario as the partner could also push the agent. To simulate this situation, the Ego network has correspondingly to be equipped with an element containing its spatial position, called "pos.Ego" in Fig. 4 (to simplify the figure, elements "grasp" and pos.candy are omitted in this and the later figures). In the following, two possible situations are considered, (1) the agent pushing the partner ("Ego push Partner") and (2) the partner pushing the agent ("Partner push Ego"). In these situations the agent may act as an actor (corresponding to a grammatical subject in an active phrase) or as a patient (corresponding to a grammatical object in an active phrase). Therefore, instead of having one unit for each individual as in the networks explained above, we introduce now two units to represent each individual, the agent and the partner. The corresponding subject units and object units are arranged under the column "subject" and "object" (Fig. 4a,b) and are connected via mutual inhibition.

To represent a verbally given situation like "Ego push Partner" in the network, some way is required to define roles. Here we assume that the item given first in time functions as

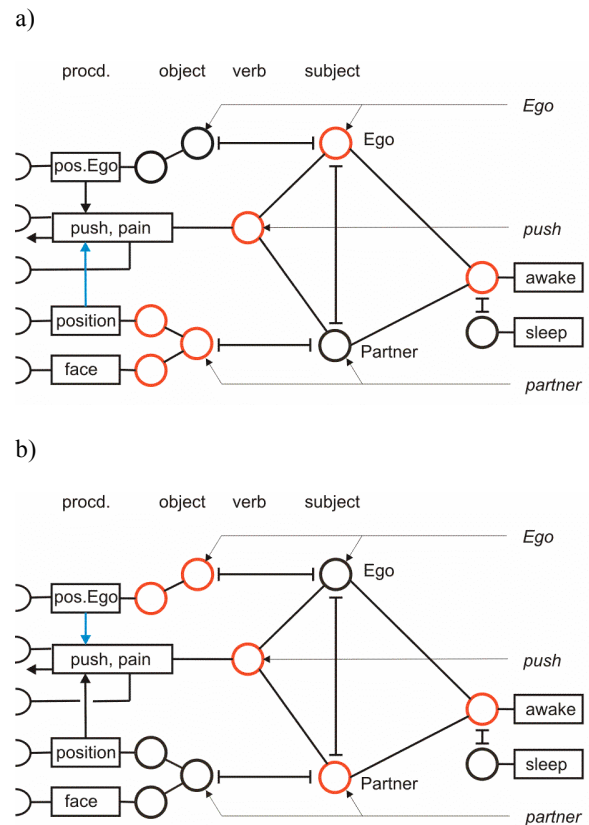


Fig. 4. A network allowing for ToM, being able to represent an egocentric view (a, situation "Ego push Partner") and the view as seen by the partner (b, situation "Partner push Ego"). Units for individuals (agent, partner) can be represented by an 'object unit' or a 'subject unit', as indicated in the top line. Sensorimotor, or procedural, networks can be found under the heading 'procd.', action units under 'verb'. For further explanation see Fig. 1 and text.

subject, the second as verb, and the third as object. The network shown in Fig. 4a,b maps the temporal order into the neuronal structure. Beginning with situation (1) input *Ego* is given first and is immediately followed by *push*. This leads to an activation of the unit Push and the Ego-subject unit (Fig. 4a, red) and an inhibition of both the Ego-object unit and the Partner-subject unit. Ego-subject unit is activated rather than the Ego-object unit because only the former is supported by activation of the unit Push. Later, both partner units will be activated via input "*partner*". As the Partner-subject unit is already inhibited, the Partner-object unit will win, in turn activating its position unit (Fig. 4a, red). Thus, all units required to represent situation (1)—the agent performs a push directed to the partner position—are active. In this way, this network can represent the egocentric view as was already possible for the networks shown in Figs. 1 and 2.<sup>1</sup>

<sup>1</sup> If the situation is not given by verbal input, but for example by visual observation, the roles of the different items actor, action and patient may

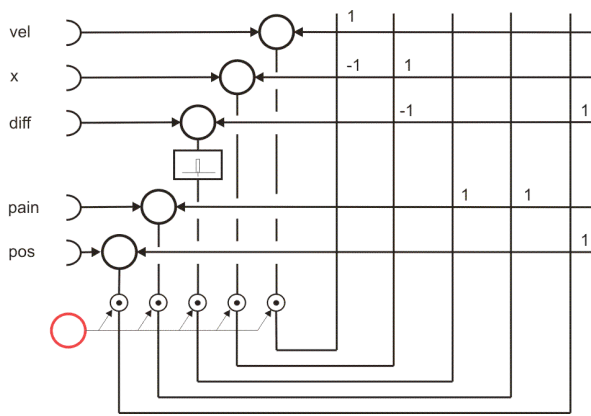


Fig. 5. A recurrent network using five IC units that shows in more detail the sensorimotor element termed “push, pain” in Fig. 5. The uppermost three units represent a simple (one-dimensional) form of the push controller (vel: velocity of the end-effector, x: position of the end-effector, also used as motor output, diff: spatial difference between actual position and goal position, the latter represented by unit “pos”). The recurrent network “pain”, consisting of one unit, when activated long enough represents the neuronal substrate for feeling pain. The unit diff possess a nonlinear activation function that allows to activate the pain network when the activation of the unit diff has approached a value of about zero. The activation of the complete network is controlled by a motivation unit (red circle).

The same network can however correspondingly represent situation (2) “Partner push Ego”. To this end, the partner units, now representing the actor, are first activated together with Push, whereas in a later step unit Ego is activated. In a corresponding way, at the end Partner-subject unit, unit Push, as well as Ego-object unit and Ego-position unit remain active (Fig. 4b).

If the agent is confronted with the latter situation “Partner push Ego” for the first time, it may suffer from a painful feeling, which will then be associated with being pushed. The network whose activation is accompanied by the subjective experience of pain (Fig. 4, box ‘push,pain’), is integrated into the push network in the following way. The pain network is activated when the controlled position of the tip of the arm reaches the goal position, the pain being associated with the goal position.

To illustrate how the networks push and pain and the input from the position network are connected, in Fig. 5 a minimal version of this subnetwork is depicted in more detail. The network altogether consists of five IC units plus one motivation unit. The push network contains three units, one representing position of the end-effector of the arm characterized by one dimension, x, the (constant) velocity of the end-effector, vel, and a unit diff representing the difference between the actual position x and the target position pos. Unit diff has a nonlinear activation function

be internally represented by different salience values provided by neuronal systems able to detect these different roles.

providing an output of 1 in a small interval around an activation value of zero, and providing a zero output otherwise. In all three cases, one unit suffices to represent the corresponding values as we focus on a one-dimensional example.<sup>2</sup> Furthermore, there is an RNN, consisting of one unit that when activated represents a painful state (pain). Unit pain is activated as soon as the end-effector meets the target position (diff = 0). We will not deal with the question how these weights are learned.

If—after this network has been installed and the situation (1) “Ego push Partner” is activated (either as active behaviour or only as imagined, i.e. simulated, behaviour)—the position of the partner will be associated with the feeling of pain (arrow highlighted in blue in Fig. 4a). In this way, our agent can simulate and thereby experience the experience of the partner without confusion between the two individuals. This means that the agent shows the ability being endowed with empathy (following the definition of Decety and Jackson, 2004: “Empathy accounts for the [...] subjective experience of similarity between the feelings expressed by self and others without losing sight of whose feelings belong to whom”). Coming back again to the second situation (Fig. 4b), “Partner push Ego”, the agent can simulate the view of the partner being an actor. Now the position of the agent is provided to the push network (in Fig. 4b depicted by a blue arrow). Therefore the network of the agent can simulate that the agent himself is receiving a push and experiencing a painful feeling. Thus, the simulated partner can now be experienced as to experience the pain.

Taken together, the agent equipped with a network as shown in Figs. 3, 4 can experience an egocentric view as was already possible for the networks shown in Fig. 1 or 2 (see Figs. 3a and 4a). In addition, the agent is able to ‘put himself into the shoes of the partner’ in two ways: the agent can try to understand the view of its partner onto objects (Fig. 3) or onto itself (Fig. 3b and 4b), i.e. “seeing himself with the eyes of the conspecific” (ToM), and can experience the experience of the partner (Fig. 4a) by simulating the feeling of the partner. The simulation of the partner is of course based on the innate and learned structures underlying his own ability to feel.

## Mutualism

A further evolutionary as well as developmental step that, according to Tomasello (2009) is unique for humans, is described by the term mutualism. Mutualism concerns the property of an agent to cooperate with another agent in such a way that both individuals perform—possibly different—actions by which a common goal should be reached and where both individuals will profit. A simple case is to carry a heavy load (e.g. to move a table around obstacles). A formally related task has to be solved by a hexapod walker where the legs are considered to be driven by independent controllers, but the legs being mechanically coupled via the body and the ground. For this problem two different solutions have been proposed. One solution possibly realized by insects exploits the mechanical coupling of the legs applying an extremely

<sup>2</sup> Note that we reduce these networks to a minimum size in order to better explain the essential aspects. Of course, each network could be expanded to consist of a large number of units without touching the basic statements made here.



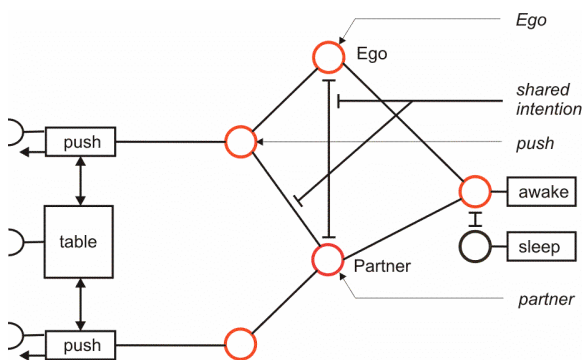


Fig. 6. A network allowing for the control of mutualistic behaviour. If input “shared intention” is activated, the (excitatory and inhibitory) connections between the subnetworks representing the agent (Ego) and the partner are interrupted. Therefore, both subnetworks can be used simultaneously to simulate actions that pursue a common goal. For further explanations see text and Fig. 1.

decentralized control structure (Schmitz et al., 2008). As an in our context more interesting alternative, Cruse and Schilling (2010) and Schilling (2011) proposed the application of an internal model that allows to simulate the legs plus their mechanical coupling through the world. Using this model each leg controller provides commands to its leg in such a way that each individual leg supports the common goal, namely moving the body forward. Applying this example to our problem of considering two independent agents able to behave mutualistically, the controller of each agent should correspondingly possess a model not only of itself, but also of the partner and the relevant environmental conditions. Together, these three elements form a ‘supermodel’. In analogy to Tomasello’s terminology, this model might also be called a “we-model”. Application of this supermodel can correspondingly be used for *probehandeln*, i.e. imagined behaviour, in order to reach a common goal. Indeed, Tomasello argued that the ability to have a we-mode is a prerequisite for developing a common goal.

What are the requirements for such a we-model to be implemented? First, the ability has to be given that actions of both the agent and the partner can be simulated independently and simultaneously. This means that it does not suffice to have only one body model that can be used to either simulate the Ego or the partner as was the case for the ‘shared-circuits’ networks shown above (Figs. 2, 3, 4). Rather, both motivation units, Ego and Partner, require access to separate behavioural elements (e.g. push) and a body model each. In Fig. 6, as in Figs. 2, 3, 4, the body model is not shown explicitly, but is graphically embedded in the push network. Both body models have to be connected via a model simulating (part of) the world to represent the actual situation, in our example the table to be carried. Furthermore, to activate the we-model, the mutual inhibition between both motivation units Ego and Partner has to be suppressed (Fig. 6). A suppression is also necessary for the connection between the motivation unit Partner and the push model which in the networks shown in Figs. 3, 4 is necessary because the latter is shared between the

Ego and the Partner network. Therefore, the we-model is activated by an input termed shared intention in Fig. 6. Tomasello has already considered shared intention a crucial property for a system showing mutualism. If this mode has been adopted, the we-model can be used to search for a solution to a given problem, for example moving the table. This search, of course, takes into account actual sensory information, e.g. position of the table relative to both individuals, movement of the other individual and possibly verbal information.

## Discussion

In our earlier work, we proposed a network that is able to control behaviour (walking, climbing, navigation) using a behaviour-based architecture and that has been expanded to show a fundamental cognitive ability, namely to be able to plan ahead. Here we propose several expansions of this network, *reaCog*. As these expansions follow the basic structure of *reaCog*, they can easily be implemented in the *reaCog* architecture. Using a typical section of *reaCog*, as an example, we start with an egocentric system (Fig. 1) that contains a body model, but is not able of mirroring. In the first step, we introduce a new connection that allows the egocentric system to apply a mirror system, i.e. to interpret behaviours observed when being performed by other individuals (Fig. 2). However, application of shared circuits alone does not appear sufficient to allow for the representation of how the world is represented by others, i.e., to allow the network shown in Fig. 2 to solve the Sally-Anne task. The latter is however possible for the networks developed in the next step (Figs. 3 and 4), which in addition contain a representation of parts of the partner’s memory. The latter concerns the position of an object, the candy in the example shown in Fig. 3 or the position of the partner (Fig. 4). In the latter example, (Figs. 4, 5), we explain in more detail how this system might implement subjective feelings which could be attributed to other individuals. Both networks are able to apply ToM. The architecture shown in Fig. 4 is still based on the application of shared circuits as the push/pain network can be connected to either the unit Ego or the unit Partner. Separation into subject units and object units is required to represent the different roles the agents have to play in this paradigm. In contrast to the egocentric systems (Figs. 1, 2), the systems depicted in Figs. 3 and 4 may be called allocentric.

Fig. 6 shows what additional connections may be required to allow for mutualism. Here two body models can be activated simultaneously and the connections allowing for sharing circuits are inhibited. Application of such a we-model is suited to allow for what often has been called the third person’s view. The step from a network as shown in Fig. 4 to that presented in Fig. 6 appears to correspond to an idea proposed by Keyers and Gazzola (2011) who draw a distinction between application of shared circuits, used for mirroring to understand the partner at a lower, intuitive, non-cognitive level, and another system involving different brain areas when subjects are asked to reflect on others. According to Keyers and Gazzola, both mechanisms are activated according to the abstraction level of the actual task. Such a two-body model appears also to be helpful to explain a number of experimental results reviewed by Sebanz et al.

(2006) and Vesper et al. (2010) which show that subjects require shared representations of tasks including the simulation of the expected behaviour of confederates.

It might be tempting to speculate that the existence of these two body models might form the basis of some illusory own-body perceptions where, due to specific neuronal deficits, subjects can experience two body representations and self-identification refers either to the physical body (Autoscopy), to the illusory body (Out-of-Body experiences) or to both either simultaneously or in alternation (Heautoscopy) as described by Blanke and Metzinger (2009). In our system such illusions may result if accidentally both body models are connected to the unit Ego, a connection not depicted in Fig. 6.

## Acknowledgements

This work has been supported by the Center of Excellence 'Cognitive Interaction Technology' (EXC 277), by the EC-IST EMICAB project # FP7 – 270182 and by a DAAD postdoctoral fellowship.

## References

- Bläsing, B. (2006). Crossing large gaps: A simulation study of stick insect behaviour. *Adaptive Behavior*, 14(3):265–285.
- Blanke, O. and Metzinger, T. (2009). Full-body illusions and minimal phenomenal selfhood. *Trends in Cognitive Sciences*, 13:7–13.
- Chalmers, D. J. (1996). *The conscious mind*. Oxford University Press, New York.
- Cooper, R.P. and Shallice, T. (2006). Hierarchical schemas and goals in the control of sequential behavior. *Psychol. Rev.* 113:887–916.
- Cruse, H. (2003). The evolution of cognition — a hypothesis. *Cognitive Science*, 27(1):135–155.
- Cruse, H. (1999). Feeling our body — the basis of cognition? *Evolution and Cognition*, 5:162–173.
- Cruse, H. (2010). The talking stick: A cognitive system in a nutshell. In Giuliani, L., editor, *Jahrbuch Wissenschaftskolleg zu Berlin*, pages 52–61. Wissenschaftskolleg, Berlin.
- Cruse, H., and Schilling, M. (2010). Getting cognitive. In Bläsing, B., Puttke, M. and Schack, T., editors, *The Neurocognition of Dance*, pages 53–74. Psychology Press, London.
- Cruse, H. and Wehner, R. (2011). No need for a cognitive map: Decentralized memory for insect navigation. *PLoS. Comp Biol.*
- Decety, J., Jackson, P.L. (2004). The functional architecture of human empathy. *Behav. Cogn. Neurosci Rev.* 3: 71–100.
- Dürr, V., Cruse, H. and Schmitz, J. (2004). Behaviour-based modelling of hexapod locomotion: Linking biology and technical application. *Arthropod Struct. Develop.* 33(3):237–250.
- Feldman, J. and Narayanan, S. (2004). Embodied meaning in a neural theory of language. *Brain and Language* 89 (2):385–392.
- Freud, S. (1911). Formulierung über die zwei Prinzipien des psychischen Geschehens. In *Gesammelte Werke*, Bd. VIII, pages 229–238.
- Gallese, V. & Lakoff, G. (2005). The brain's concepts: the role of the sensory-motor system in conceptual knowledge. *Cognitive Neuropsychology* 22(3–4):455–479.
- Ishida, H., Nakajima, K., Inase, M., Murata, A. (2010). Shared mapping of own and others' bodies in visuotactile bimodal area of monkey parietal cortex. *Journal of Cognitive Neuroscience* 22:83–96.
- Jeannerod, M. (2006). *Motor Cognition — What Action tells the Self*. Oxford: University Press.
- Jeannerod, M. (1999). To act or not to act: Perspectives on the representation of actions. *Quarterly Journal of Experimental Psychology*, 52A:1–29.
- Keysers, C. and Gazzola, V. (2011). Integrating simulation and theory of mind: from self to social cognition. *Trends in Cognitive Sciences*
- Kilner, J.M., Paulignan, Y. and Blakemore, S.J. (2003). An Interference Effect of Observed Biological Movement on Action. *Current Biology* 13:522–525.
- Kühn, S., and Cruse, H. (2007). Modelling memory functions with recurrent neural networks consisting of input compensation units: II. Dynamic situations. *Biological Cybernetics*, 96(5):471–486.
- Makarov, V.A., Song, Y., Velarde, M.G., Hübner, D. and Cruse, H. (2008). Elements for a general memory structure: properties of recurrent neural networks used to form situation models. *Biological Cybernetics* 98(5):371–395.
- McFarland, D., Bösner, T. (1993) *Intelligent behavior in animals and robots*. MIT Press, Cambridge, MA
- Menzel, R., Brembs, B. and Giurfa, M. (2007). Cognition in Invertebrates. In Kaas, J.H., editor, *Evolution of Nervous Systems, Vol. II: Evolution of Nervous Systems in Invertebrates*, chapter 1.26, pages 403–422. Academic Press, Oxford.
- Narayanan, S. (1997). Talking the talk is like walking the walk: A computational model of verbal aspect. In COGSCI-97, pages 548–553. Stanford, CA.
- Rizzolatti, G. and Luppino, G. (2001). The cortical motor system. *Neuron* 31: 889–901.
- Schilling, M. (2011). Universally manipulable body models — dual quaternion representations in layered and dynamic MMCs. *Autonomous Robots* 30(4):399–425.
- Schilling, M. and Cruse, H. (submitted). Cognition as recruitment of reactive systems.
- Schilling, M. and Spranger, M. (2010), "Embodied posture verbs: Emergence of a vocabulary for describing postures", Conference on Conceptual structure, discourse and language (CSDL) & Embodied and situated language processing (ESLP) 2010, San Diego.
- Schmitz, J., Schneider, A., Schilling, M. and Cruse, H. (2008). No need for a body model: Positive velocity feedback for the control of an 18-DOF robot walker. *Applied Bionics and Biomechanics*, 5(3):135–147.
- Sebanz, N., Bekkering, H., Knoblich, G. (2006). Joint action: Bodies and minds moving together. *Trends in Cognitive Sciences* 10:70–76.
- Spranger, M., Höfer, S. and Hild, M. (2009). Biologically inspired posture recognition and posture change detection for humanoid robots. In *Proceedings of ROBIO'09: IEEE International Conference on Robotics and Biomimetics*, pages 562–567.
- Steels, L. (1995). A self-organizing spatial vocabulary. *Artificial Life*, 2(3):319–332.
- Steels, L. (2003). Intelligence with representation. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences* 361 (1811):2381–2395.
- Steels, L. and Spranger, M. (2008). The robot in the mirror. *Connection Science*, 20(4):337–358.
- Tomasello, M. (2009). *Why we cooperate*. Boston Review Book, MIT Press, MA.
- Vesper, C., Butterfill, S., Knoblich, G., & Sebanz, N. (2010). A minimal architecture for joint action. *Neural Networks*, 23, 998–1003.
- Wehner, R. (2008) The desert ant's navigational toolkit: procedural rather than positional knowledge. *Navigation* 55(2):101–114.

This document was last revised on June 6, 2011.

# Artificial Gene Regulatory Networks and Spatial Computation: A Case Study

Sylvain Cussat-Blanc<sup>1</sup>, Nicolas Bredeche<sup>2</sup>, Hervé Luga<sup>1</sup>, Yves Duthen<sup>1</sup> and Marc Schoenauer<sup>2</sup>

<sup>1</sup> University of Toulouse - IRIT; 2 rue du Doyen Gabriel Marty 31042 Toulouse, France  
{cussat; luga; duthen}@irit.fr

<sup>2</sup> TAO/LRI; Univ Paris-Sud - CNRS - INRIA Saclay IdF; F-91405, France  
{bredeche; marc}@lri.fr

## Abstract

This paper explores temporal and spatial dynamics of a population of Genetic Regulatory Networks (GRN). In order to so, a GRN model is spatially distributed to solve a multi-cellular Artificial Embryogeny problem, and Evolutionary Computation is used to optimize the developmental sequences. An in-depth analysis is provided and show that such a population of GRN display strong spatial synchronization as well as various kind of behavioral patterns, ranging from smooth diffusion to abrupt transition patterns.

## Introduction

Widely studied in Biology, Gene Regulatory Networks (GRN) have drawn in recent years a growing attention from the field of Artificial Life and Evolutionary Computation. Indeed, GRN are known to display rich dynamics and have been both experimentally studied through simplified models (Jakobi, 1995; Banzhaf, 2003) as well as applied to control optimization problems such as the well-known inverted pole balancing problem (Nicolau et al., 2010) and foraging agents (Joachimczak and Wróbel, 2010). In these recent works, evolving artificial GRN have always been shown to be competitive with the state-of-the-art neuro-evolution techniques, possibly because of rich internal dynamics. However, while temporal dynamics within a single GRN have already been studied Banzhaf (2003), the spatial dynamics resulting from coupling of several GRNs remains to be explored.

The core motivation in this paper is to describe and study such temporal *and* spatial dynamics of a population of GRN in the context of a spatial computation problem. The methodology followed relies on Evolutionary Computation to provide optimization tools so as to fine tune the GRN parameters and structure for solving a typical multi-cellular artificial embryogeny problem. In this setup, the GRNs act as a decision model that is spatially distributed over a set of *cells* that interact on a local basis such that the whole *organism* converges towards a global state that is the closest possible to a pre-defined target state (e.g. a particular pattern).

Rather than performance on target matching, we study the emerging spatial and temporal dynamics during the course

of the developmental process from the initial state to the end of development. Experimental investigations show that gene expressions are indeed strongly synchronized among GRNs, and display several behavioral patterns from smooth diffusion to abrupt transitions.

In the following, a review of existing artificial GRN models is provided. Then, the GRN model originally proposed by Banzhaf (2003) is introduced as well as the developmental model used in this study. The combination of both models is described, and experimental investigations are conducted on the spatial and temporal dynamics of GRN. The paper concludes with a discussion and sketches future directions.

## Background on artificial regulatory networks

Many current developmental models rely on an Artificial GRN to simulate cell differentiation. These systems are more or less inspired by gene regulation systems of living systems. In living systems, organisms' cells have several functions. They are described in the organism genome and their expressions are controlled by the regulatory network (Davidson, 2006). Cells use external signals from their environment to activate or inhibit the transcription of genes into mRNA (messenger RiboNucleic Acid), the copy of the daughter cell's DNA (DeoxyriboNucleic Acid). Cells collect external signals through protein sensors localized on the cell membrane. Then, gene expression within a cell determines its behavior.

Eggenberger (1997) was one of the first to use a regulatory network to generate a 3-D organisms able to move in its environment by modifying its morphology. Reil (1999) proposed a biologically plausible model, with a genome defined as a vector of numbers. In this model, each gene starts with a particular sequence (0101), named the "promoter". Then, a graph visualisation is used to observe gene activations and inhibitions over time with randomly generated networks. Observations revealed the existence of several patterns such as gene activation sequencing, chaotic expressions or cyclic expressions. The author also pointed out that the system was able to display pattern self-repairing after random genome deteriorations. Banzhaf (2003) also described an artificial



GRN model strongly inspired by real-world gene regulation. This model will be detailed in the next section.

Starting from these two seminal models, various extensions and variations have been explored, for addressing various concerns and applications. Several works addressed Artificial Embryogeny problems with models of GRN ranging from cellular automaton modeling (Chavoya and Duthen, 2008) to stripped-down version of GRN combined with complex developmental systems (Knabe et al., 2008; Joachimczak and Wróbel, 2008; Doursat, 2008). Some works have also addressed control problems: using GRN as a control function to map a virtual robot's sensory inputs to its motor actuator values. This has been applied in various setup, from foraging agents (Joachimczak and Wróbel, 2010) to pole balancing (Nicolau et al., 2010).

Few case studies have been done to explain how regulatory networks can solve these problems. Schramm et al. (2010) studies the impact of the evolutionary process on the network itself. Other papers of the literature such as Mjølness et al. (1991) or Thomas et al. (1995) propose an analysis of the regulatory network dynamics in a biological point of view. However, few papers deal with the analysis of such dynamics on artificial regulatory networks, which could be useful if we want to use effectively the computational abilities of these models. The aim of this paper is to show the gene expression temporal answer of a regulatory network to solve a spatial problem. For this purpose, we use Banzhaf's GRN (Banzhaf, 2003) and its extension to a computational model presented in (Nicolau et al., 2010). The next section describes this model.

## The gene regulatory network

### The model

In this work, we consider the artificial Gene Regulatory Network (GRN) introduced by Banzhaf (2003). In this model, the network is coded into the genome as a sequence of 32-bit strings (termed *sites*). Each gene in the genome is marked by a particular sequence named the "promoter". When a promoter is detected, the next five sites represent a gene sequence that codes for a protein to be produced. Each site codes for a different molecule of the protein. The concentration of this protein will determine the expression level of the corresponding gene.

To determine the protein's concentration and thus the gene expression level, two sites, coded upstream of the promoter, enhance and inhibit the protein production. The dynamics of enhancer signal  $e_i$  and inhibitor signal  $h_i$  of a protein  $i$  are given by the following equations:

$$e_i = \frac{1}{N} \sum_{j=1}^N c_j \exp^{\beta(u_j^+ - u_{max}^+)} \quad (1)$$

$$h_i = \frac{1}{N} \sum_{j=1}^N c_j \exp^{\beta(u_j^- - u_{max}^-)} \quad (2)$$

where  $N$  is the total number of proteins,  $c_j$  is the concentration of the protein  $j$ ,  $\beta$  is a scaling factor,  $u_j^+$  (resp.  $u_j^-$ ) is the matching degree of the enhancer (resp. inhibitor) site with the protein  $j$  and  $u_{max}^+$  (resp.  $u_{max}^-$ ) is maximum enhancer's (resp. inhibitor's) matching degree observed in the whole genome. The matching degree  $u_j^+$  (resp.  $u_j^-$ ) consists in counting the number of "1" resulting from the application of a XOR operation to the protein  $j$  and the enhancer (resp. inhibitor) pattern. The exponential function increases the impact of high value of gene expression and filter low values.

Finally, the concentration of produced protein  $p_i$  follows the differential equation  $dc_i/dt = \delta(e_i - h_i)c_i - \Phi(1.0)$ , where  $\delta$  is a scaling factor and  $\Phi(1.0)$  constrains the sum of all concentration equals to 1.0.

### Extension to a computational model

Originally, Banzhaf's artificial GRN is limited to study internal network dynamics. In order to use this model as a control function, Nicolau et al. (2010) proposed an extension by adding inputs and outputs to the regulatory network. This extension is detailed in the following.

**Inputs** Input values are coded with integers that will correspond to existing proteins. These input proteins can be involved in the regulatory process in two different ways: with their signatures to be considered during the matching process (in equations of  $e_i$  and  $h_i$ ) or with their input value to modify the differential equation  $dc_i/dt$  of protein concentrations. Here, the second solution has been chosen as it allows a better resolution with regard to a continuous domain of the problem addressed in this paper.

**Outputs** In order to produce outputs in the regulatory networks, genes are separated into classes: transcription factors *TF-genes* and product proteins *P-genes*. Whereas TF-genes play the roles of regulatory proteins as in the original Banzhaf's model, P-genes are only regulated but do not regulate other proteins: their expression levels provide the desired output signals. These two kinds of genes are identified by introducing two new promoters, whose signatures are chosen so that their probability of occurrence is equivalent and their matching as low as possible.

In the following, the regulatory network is used to produce cell differentiation, expressed by a cell coloration, while the developmental model described in the next section is responsible for the generation of the shape.

### The developmental model

The Generative Developmental System (GDS) *Cell2Organ* is composed of three layers of simulation: a *chemical* layer,

a *hydrodynamic* layer and a *physical* layer. These three layers can be enabled or disabled according to the needs of the experimentation. In the scope of this work, only the chemical layer is considered and will be described. More details about the developmental model are given in (Cussat-Blanc et al., 2008, 2010b,a).

The environment, implemented as a 2-D toroidal grid, contains several kinds of substrates. They spread within the grid, minimizing the variation of substrate quantities between two neighboring points. These substrates can spread on the grid at different speeds. Substrates can interact together in order to simulate a simplified chemical reaction. Only cells can trigger substrate transformations and collect or consume the energy of the transformation.

Cells act in the environment. Each cell contains sensors and has different abilities (or actions). An action has a energetic cost for the cell that will trigger it. An action selection system allows the cell to select the best action to perform at any moment of the simulation. This system is based on a set of rules *precondition*→*action* (*priority*). It uses data given by sensors to select the best action to perform.

*Division* is a particular action that can be performed if three conditions are respected. First, the cell must have at least one free neighbor to create the new cell. Secondly, the cell must have enough vital energy to perform the division (this required level is defined *a priori*). Finally, during the environment modeling, additional conditions can be added. A new cell created after division is totally independent and interacts with the environment. During the division, the GRN is executed in order to determinate the cell's color according to the morphogen quantity observed by the cell.

This model has been applied to shape generation (assembly of cells) in (Cussat-Blanc et al., 2008): a simple control function is evolutionary optimized to control cells so that it is possible to produce target shapes at the level of the organism within an environment with pre-positioned morphogens. In the current work, the control function considered is the extended Banzhaf's GRN model, coupled with an Evolution Strategies optimizer. Coupling the two models (GRN and developmental) is described in the next section.

## Coupling of the GRN and the GDS

### Precomputation of the cell differentiation

Different morphogen gradients are added to position cells in the environment. These morphogens are dedicated to differentiation. The configuration of these gradients will be described precisely for each experiment.

The cell differentiation is represented in the developmental model by a cell coloration. The concentration in morphogens measured by the cell in the environment defines the inputs of the regulatory network. These concentration are scaled to the range  $[0.0, 0.3]$  in order not to overload the production of other regulatory proteins (the sum of all concentrations is normalized in the range  $[0.0, 1.0]$ ). To obtain

the cell coloration, each cell executes the regulatory network during its division stage. Only one color can be expressed. Therefore, the maximum of the expression level of all genes is taken after a stabilization of the network (chosen empirically after 1000 time steps of the regulatory network evolution). This gene expression will finally give the cell color during the development of the organism.

Because the cell can be positioned in a coordinate system and the morphogen gradients are prepositioned, the differentiation mechanism can be precomputed before the development stage. In other words, the problem can be translated to the search of an integer matrix. Each value of the matrix corresponds to the color of the corresponding cell in the chemical environment (1 for white, 2 for red and 3 for blue). The same regulatory network is independently executed at each point of the matrix with the morphogen concentrations that corresponds in the chemical environment. The regulatory network is used to generate a differentiation matrix that correspond to the desired pattern (also translated to an integer matrix). The developmental model then determines cell coloration using this differentiation matrix during the organism growth. During temporal development, this matrix thus simplifies computation within the model as cell differentiation can be directly set at cell creation. This is justified in the present context as pre-computing morphogen diffusion is a sub-problem that may not be critical for studying the already rich GRN dynamics.

### Evolutionary algorithm

A classical (250+250) evolution strategy (ES) evolves a population of regulatory networks coded by the binary string previously presented. The (250+250) evolution strategy consists in producing 250 offsprings from 250 parents and choosing the 250 best genomes to form the next population. The fitness function that evaluates each genome consists of counting the number of cells that do not match the desire pattern (wrong cell coloration). The evolution strategy is launched for 100 generation to minimize the quadratic error. In the following, the error is computed as the difference for each pixels between the image generated by the organism (cell differentiation determines pixel color) and the target image.

Genome modifications are only regulated by a common bit-flip mutation operator. The mutation rate is set to 2% at the beginning of the run and adapted by the 1/5 rule of evolution strategies (Rechenberg, 1994): (1) the mutation rate is doubled when the rate of successful mutation is higher than 20%; (2) the mutation rate is divided by two when the rate of successful mutation is lower than 20%; (3) the mutation rate is doubled when the number of gene mutations in the population is less than 250 by generation.

The regulatory network's genome is randomly initialized. It is then duplicated 9 times with a mutation rate of 2% in order to increase the appearance probability of reg-

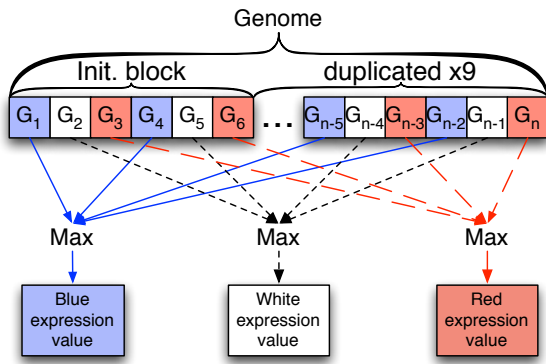


Figure 1: The genes of the genome are classified into three sub-parts: blue, white and red genes. The final expression value of each color is given by the highest value of the corresponding genes.

ulation sites. However, only three genes are necessary to code the three needed colors (blue, white and red cell colors). The duplication of the genome implies a strong possibility to have more than the three needed genes coded in the genome. As described on figure 1, the genome is divided in three sub-part. Each part codes for a specific color: blue, white and red. The highest gene expression value in one of the three sub-parts of the genome is taken as the expression value of the corresponding color.

Each differentiation matrix is developed only one time because the problem is deterministic. In other words, a regulatory network will always generate the same differentiation matrix and thus the same cellular pattern.

Figure 2 presents the convergence curves of the evolution strategy applied to our two problems of flag development presented in the next section. We can observe a stepwise evolution due to the only use of mutation. Moreover, even if the algorithm is set for 100 generations, it converges much faster (approx. 30 generations).

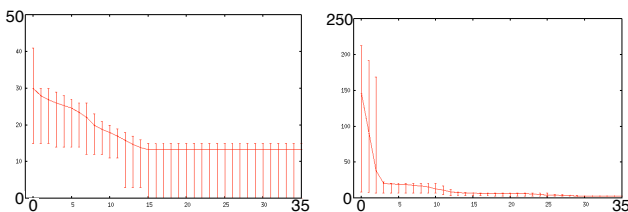


Figure 2: Convergence of the ES applied to a 45 cells French flag (left) and a 213 cell Japanese flag (right). X-axis represents the generation and the ordinate the min, mean and max fitness values (number of errors) for each generation.

## Experiments

### Benchmark: the French flag problem

In recent years, the French flag problem has become a classical benchmark for evolutionary computation. Introduced by Wolpert at the end of the 1960s (Wolpert, 1968), it consists in developing a French flag pattern starting from a single cell in the centre. This pattern is composed of three colored strips (blue, white and red). The French flag problem has various point of interests. In this paper, it is relevant as a spatial problem as it can highlight the differentiation capacities of a GRN-controlled developmental model: the color changes in the flag can easily be interpreted as a functional switch of the cell.

This benchmark has been addressed using various approaches. Lindenmayer (1971) used it to point out the capacity of his L-Systems to generate predefined shapes. Miller (2003) used a cartesian genetic programming approach and addressed self-repairing issues. Bowers (2005) used an embryonic developmental model to produce a French flag. (Devert, 2009) addressed this problem with various methods based on using the NEAT neuro-evolution method (Stanley, 2004), Jaeger's Echo State Networks (Jaeger, 2001) and a reaction-diffusion model bearing resemblance with the original Miller's model.

This benchmark became quite famous in the Artificial GRN community as it can be used to show gene expressions of cells (Banzhaf, 2003; Knabe et al., 2008; Joachimczak and Wróbel, 2008). The major difference with previous work is that our contribution emphasizes the analysis on internal dynamics rather than focusing on pure performance and generalization. To this end, the problem is briefly described and experimental results are analysed, with a particular emphasis on internal dynamics of GRN as well as the spatial resolution of the problem in terms of gene expressions.

### Relationship between spatiality and temporality

Two different target shapes are considered: a French flag (three vertical strips) and a Japanese flag (white background with a red centered circle), each with its specific properties regarding the possible impact of morphogen gradients on the GRN expression levels.

**The French flag** In this problem, two morphogen gradients are positioned horizontally and vertically. They allow a precise positioning of the cells in the environment on the x-axis and y-axis. However, the target flag is developed in the diagonal of the environment. It implies an adaptation of the regulatory network to utilize both morphogens.

The regulatory network is trained on a 9x5 flag (45 cells). The target flag is composed of 3 strips of the same size: a blue in the bottom left of the environment, a white in the center and a red in the top right part. Figure 3 shows the obtained result. The resulting image perfectly matches with

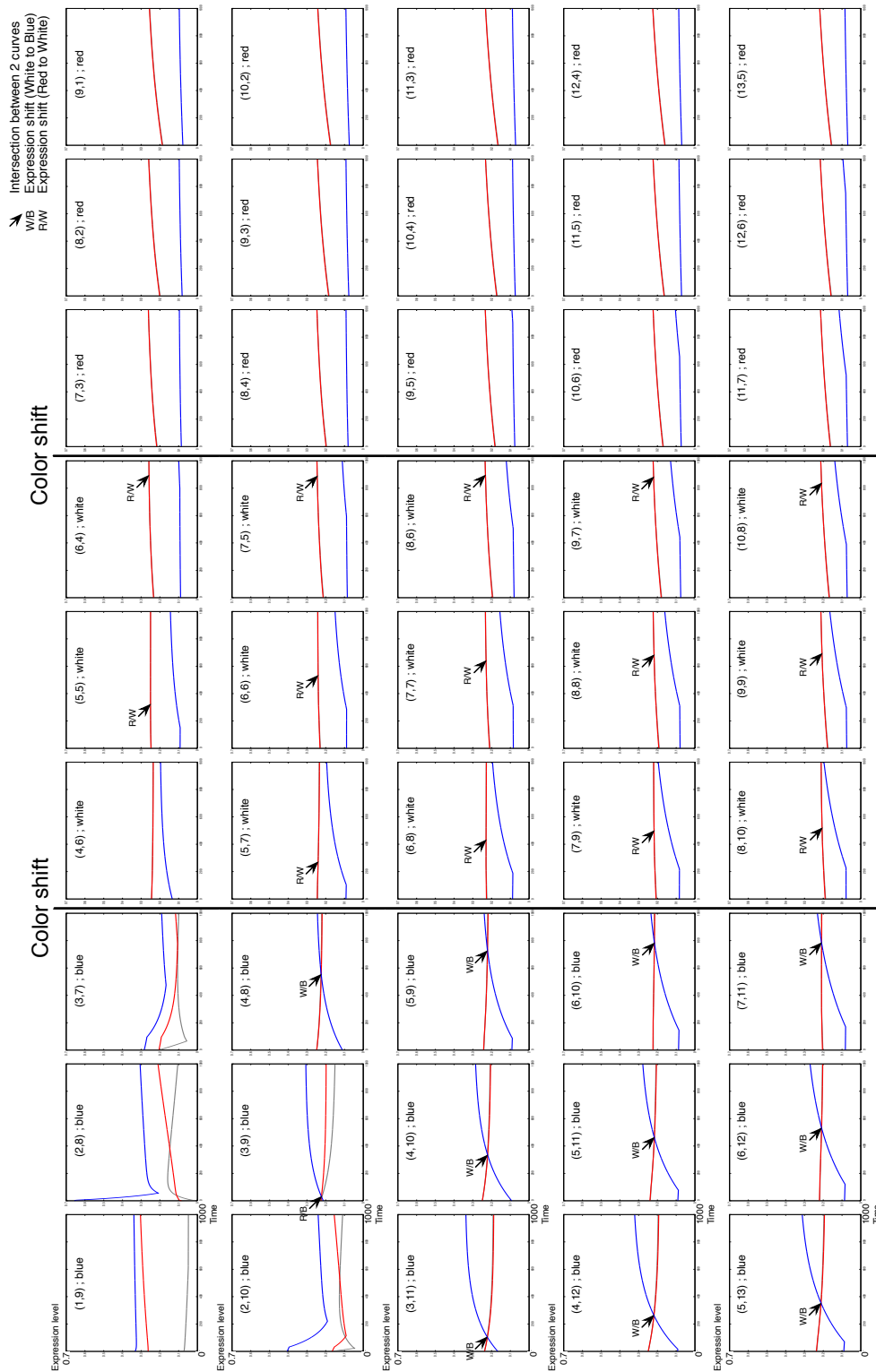


Figure 4: Variation of the gene expression levels over time for each cell of the organism. The curves correspond to the regulatory network activity of each cell of the French flag. The coordinate and the color of the cell are given by the title of each curve. We can observe a strong link between the delay of expression of appropriate gene and the distance to the color shift: the longer the distance to the color shift point, the faster the gene expression.

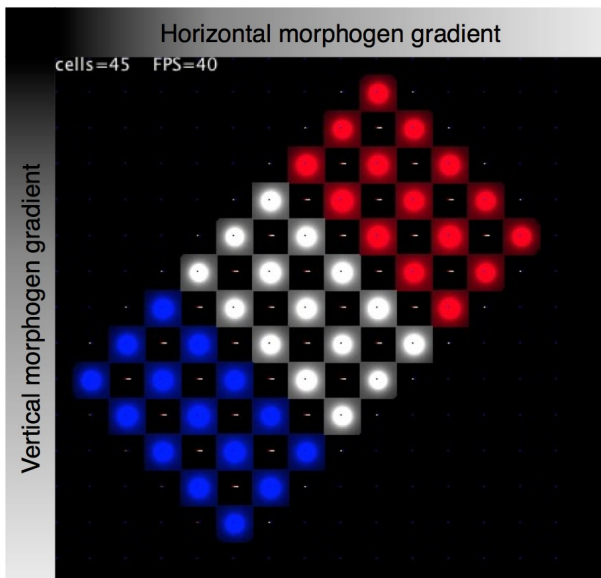


Figure 3: Development of the French flag

the target flag. To study the spatialization of the regulatory network, we extract all the curves of the color expression level over time of the regulatory network expressed in each 45 cell of the organism. These curves are presented in figure 4. The top left curve matches with the left corner blue cell of the organism in figure 3.

All these curves represent the variation of the three gene expression levels (blue, white and red) on the y-axis (scaled between 0 and 1) during the one thousand time steps of regulatory network's evolution. In the top left part of the figure, both morphogen orientations are represented according to the organism orientation.

It is interesting to notice the progressive softening of the blue curve in all curves and, at the opposite, the progressive increasing of the two other curves (red and white are almost overlapped). On the one hand, the transition between the blue curve and the white/red curves is very visible. On the other hand, the transition between white and red is hugely more smoothy. Both curves are very close all the time, except in the 5 top left curves. This exception is certainly due to a strong regulation shift in the regulatory network.

More relevant, the temporality of the color expression shifts is very observable. Considering only the blue and the white strips, the expression of the blue color is visible later and later in the regulatory network as the cell is closer to the white area.

The blue/white shift disappears from the curve when the cell must be white but we can assume by interpolation of the curves that the shift happens later. The same phenomenon is also present between the white and the red strips, as pointed out by the *R/W* black arrows. It exhibits the strong link between the temporality of the gene expression and the spatial-

ity of the problem provided by the morphogen gradients.

Figure 5 presents the extraction of the regulatory network of the best evolved candidate. The nodes represent two groups of genes: the regulation genes named G1 to G39 and the product genes (that will produce the color of the cell) named P1 to P99. The size of each node is proportional to its number of links. The architecture of this network is interesting to observe. First, almost all the genes are used. Only two genes (G23 and G33) are not linked to the regulatory network. It shows the total use of the genome and the complexity of the network extracted. Secondly, six genes (G5, G14, G16, G27, G28 and G38) are interfacing the regulatory network and all product genes except P2, which is directly linked to the regulatory network. The interface has not been coded in the network. It only emerged thanks to the evolutionary process. Lastly, in the regulatory area, three genes (G4, G25 and G26) play a central role and they are strongly connected to the rest of the regulatory area. This regulatory area is very complex with a lot of links between all the nodes. This complexity is due to the necessity to exploit both gradients (horizontal and vertical).

**The Japanese flag** In order to investigate the independence of the coordinate system to the temporality answer of the regulatory network, development of a japanese flag in a radial coordinate system is studied. The goal is thus to develop into an image with a red circle in the center of a white 13x9 rectangular shape (a total of 213 cells). The same three genes have been kept in order to establish the capacity of the GRN to switch off a particular gene.

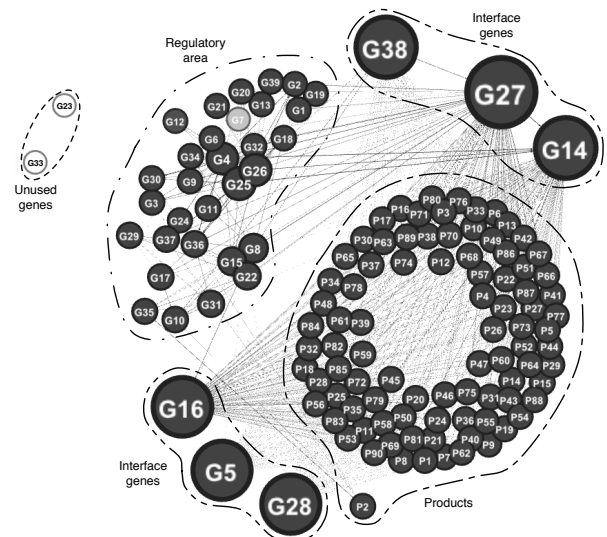


Figure 5: Gene regulatory network extracted from the best genome of the French flag with a threshold value of 19. G-genes represent regulation genes and P-genes represent the products of the regulatory network (a color expression).

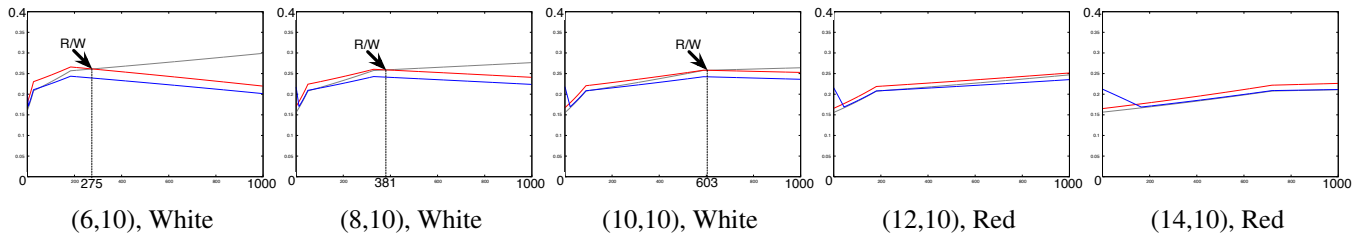


Figure 6: Gene expression level curves of 5 cells of the Japanese flag's central line. The curves' legends indicate the coordinate and the color of the cell that correspond to the gene expression.

As previously, only 15 generations is required to obtain a near-perfect flag (with only 3 pixels wrong). Figure 7 illustrates the flag obtained.

As for the previously presented French flag, all the curves of the gene regulation have been extracted in order to study the link between the temporality of the regulation and the spacialization of the problem. Figure 6 shows the curves of gene expression levels of five cells of central line: 3 whites cells and 2 red.

We can observe that all the expression levels are very close (y-axis is zoomed on the interval  $[0, 0.4]$ ). The blue gene is also very strongly expressed even if not needed in this flag. Its inhibition by the regulatory network is correctly made but seems to be very weak. The same link between the temporality and the distance to the shift is also observable as on the French flag: the closer the colors shift, the later the gene expression levels shift. The same behavior is observable elsewhere on the flag and each transition stage can be obtained by rotation.

## Conclusions and Perspectives

The goal of this work was to investigate the use of Artificial GRN in the context of a spatial problem. We combined Banzhaf's GRN model to our own developmental model *Cell2Organ*, and experimental studies have been conducted on variations of the multi-cellular flag problem, a well-known benchmark in Artificial Embryogeny. Results from

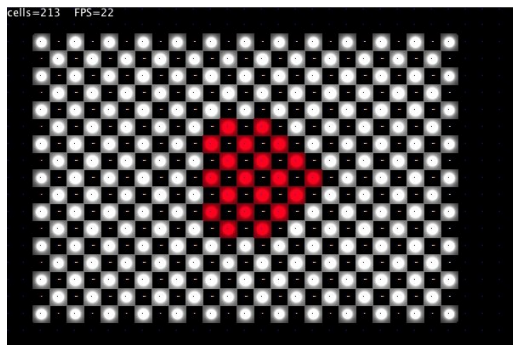


Figure 7: Development of a Japanese flag with a radial gradient.

the experiments confirm the strong link between the temporality of the gene expressions in the regulatory network and spatial parameters of the problem. Indeed, change in the cell differentiation process among the organism is correlated by significant shifting in the GRN dynamics. The temporal aspect observed here also raises numerous question regarding the ability of a population of GRN to actually generate some desired behaviors. How many steps does it take to produce a correct output? What is the expressivity of such a system, in particular, how many basin of attractions can be encoded within one GRN template? Is it possible to have, depending on the context at hand, either a fast or a smooth shift between two regimes? These questions are of particular interest to explore further GRN-based control optimization problems (Joachimczak and Wróbel, 2010; Nicolau et al., 2010).

The complexity of the regulatory network obtained was also somewhat surprising and raises the question as to the evolvability of such a representation. The regulatory network needed a large number of P-genes (not restrained in these experiments) in order to find a solution to the problem. This may be a symptom of code bloat, a well-known problem of uncontrolled growth in variable length representations and definitely requires further studies, with possible investigations with respect to penalizing bloat without undermining the model's performance.

Lastly, spatial problems addressed here are relevant for this kind of detailed study, but have limited applications in the current form. However, the field of applications is large and examples from Biology give a good indications on the variety of problems to be addressed: cell differentiation into neurones, development of muscular cell, tissues, etc. In the context of computer modelling, understanding the intrinsic properties of GRN may be relevant in a variety of problems requiring complex temporal and spatial interactions. Indeed, because of their structure, regulatory networks could be more suitable for continuous problems than other behavior controllers such as artificial neural networks or classifier systems.

## Acknowledgements

The authors would like to thank Miguel Nicolau for his help regarding the implementation of the artificial GRN model.



## References

- Banzhaf, W. (2003). Artificial regulatory networks and genetic programming. *Genetic Programming Theory and Practice*, pages 43–62.
- Bowers, C. (2005). Simulating evolution with a computational model of embryogeny: Obtaining robustness from evolved individuals. *Lecture notes in computer science*, 3630:149.
- Chavoya, A. and Duthen, Y. (2008). A cell pattern generation model based on an extended artificial regulatory network. *Biosystems*, 94(1-2):95–101.
- Cussat-Blanc, S., Pascalie, J., Luga, H., and Duthen, Y. (2010a). Morphogen positioning by the means of a hydrodynamic engine. In *Artificial Life XII*, pages 118–125. MIT Press, Cambridge, MA.
- Cussat-Blanc, S., Luga, H., and Duthen, Y. (2008). From single cell to simple creature morphology and metabolism. In *Artificial Life XI*, pages 134–141. MIT Press, Cambridge, MA.
- Cussat-Blanc, S., Pascalie, J., Luga, H., and Duthen, Y. (2010b). Three simulators for growing artificial creatures. In *Proceedings of the IEEE Congress on Evolutionary Computation (IEEE CEC 2010)*.
- Davidson, E. H. (2006). The regulatory genome: gene regulatory networks in development and evolution. *Academic Press*.
- Devert, A. (2009). *Building processes optimization: Toward an artificial ontogeny based approach*. PhD thesis, Université Paris-Sud 11.
- Doursat, R. (2008). Organically grown architectures: Creating decentralized, autonomous systems by embryomorphic engineering. *Organic Computing*, pages 167–200.
- Eggenberger, P. (1997). Evolving morphologies of simulated 3d organisms based on differential gene expression. In *Proceedings of the Fourth European Conference on Artificial Life*, pages 205–213. MIT Press Cambridge, MA.
- Jaeger, H. (2001). The echo state approach to analysing and training recurrent neural networks. *submitted for publication*.
- Jakobi, N. (1995). Harnessing morphogenesis. In *International Conference on Information Processing in Cells and Tissues*.
- Joachimczak, M. and Wróbel, B. (2008). Evo-devo in silico: a model of a gene network regulating multicellular development in 3d space with artificial physics. In *Proceedings of the 11th International Conference on Artificial Life*, pages 297–304. MIT Press.
- Joachimczak, M. and Wróbel, B. (2010). Evolving Gene Regulatory Networks for Real Time Control of Foraging Behaviours. In *Proceedings of the 12th International Conference on Artificial Life*.
- Knabe, J., Schilstra, M., and Nehaniv, C. (2008). Evolution and morphogenesis of differentiated multicellular organisms: autonomously generated diffusion gradients for positional information. *Artificial Life XI*, 11:321.
- Lindenmayer, A. (1971). Developmental systems without cellular interactions, their languages and grammars. *Journal of Theoretical Biology*, 30(3):455.
- Miller, J. (2003). Evolving Developmental Programs for Adaptation, Morphogenesis, and Self-Repair. In *Advances in Artificial Life*, pages 256–265. Springer.
- Mjolsness, E., Sharp, D., and Reinitz, J. (1991). A connectionist model of development. *Journal of theoretical Biology*, 152(4):429–453.
- Nicolau, M., Schoenauer, M., and Banzhaf, W. (2010). Evolving genes to balance a pole. *Genetic Programming*, pages 196–207.
- Rechenberg, I. (1994). Evolution strategy. *Computational Intelligence: Imitating Life*, pages 147–159.
- Reil, T. (1999). Dynamics of gene expression in an artificial genome - implications for biological and artificial ontogeny. *Advances in Artificial Life*, pages 457–466.
- Schramm, L., Valente Martins, V., Jin, Y., and Sendhoff, B. (2010). Analysis of Gene Regulatory Network Motifs in Evolutionary Development of Multicellular Organisms. In *Proceedings of the 12th International Conference on Artificial Life*.
- Stanley, K. (2004). *Efficient evolution of neural networks through complexification*. PhD thesis, The University of Texas at Austin.
- Thomas, R., Thieffry, D., and Kaufman, M. (1995). Dynamical behaviour of biological regulatory networks. biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bulletin of Mathematical Biology*, 57:247–276. 10.1007/BF02460618.
- Wolpert, L. (1968). The french flag problem: A contribution to the discussion on pattern development and regulation. *Towards a theoretical biology*, 1:125–133.



# Grounding Synthetic Knowledge:

## An epistemological framework and criteria of relevance for the scientific exploration of life, affect and social cognition

Luisa Damiano<sup>1</sup>, Antoine Hiolle<sup>1</sup> and Lola Cañamero<sup>1</sup>

<sup>1</sup>Adaptive Systems Research Group, School of Computer Science & STRI, University of Hertfordshire, College Lane, Hatfield, Herts, AL10 9AB, UK  
luisa.damiano@gmail.com

### Abstract

In what ways can artificial life contribute to the scientific exploration of cognitive, affective and social processes? In what sense can synthetic models be relevant for the advancement of behavioral and cognitive sciences? This article addresses these questions by way of a case study — an interdisciplinary cooperation between developmental robotics and developmental psychology in the exploration of attachment bonds. Its main aim is to show how the synthetic study of cognition, as well as the synthetic study of life, can find in autopoietic cognitive biology more than a theory useful to inspire the synthetic modelling of the processes under inquiry. We argue that autopoiesis offers, not only to artificial life, but also to the behavioural and social sciences, an epistemological framework able to generate general criteria of relevance for synthetic models of living and cognitive processes. By “criteria of relevance” we mean criteria (a) valuable for the three main branches of artificial life (soft, hard, and wet) and (b) useful for determining the significance of the models each branch produces for the scientific exploration of life and cognition. On the basis of these criteria and their application to the case study presented, this article defines a range of different ways that synthetic, and particularly autopoiesis-based models, can be relevant to the inquiries of biological, behavioural and cognitive sciences.

### Introduction

In his seminal article of 1989, Christopher Langton introduces the “synthetic approach” (SA) as the methodology proper to artificial life (AL) — to “put living things together”, “rather than take [them] apart” (Langton 1989, p.40). The methodological agenda that he proposes extends the focus of biological research to what is missing in the analytic approach traditionally applied to living systems. However, the plan does not merely consist in extending the focus from individual to relational components’ properties; from matter to organization; from centralised mechanisms to distributed dynamics of self-organization, as scheduled by other 20<sup>th</sup> century biological research programs. Distinctively, AL’s SA further aims to enlarge biology’s perspective from terrestrial to alternative “made by man” forms of life, and to actually include in biological heuristics, besides the question “how does it work?”, the question “why this and not that?” The intent is to investigate the essential principles of life, and to deal with the main issues about life, by attempting to “recreate” living things and their phenomenology. In other words: building artificially embodied and situated models of living systems and

phenomena in order to explore, through experimental manipulation, aspects of life usually not accessible in natural systems and scenarios.

This foundational methodological plan still unifies the three main branches of AL developed over the last two decades, since “soft”, “hard”, and “wet” AL (Bedau, 2003, p. 505), in spite of their divergent methods, all continue to refer to the SA as their basic methodology, and tend to agree in its characterisation. In addition, all of them emphasise the genuinely scientific aspiration of this methodology, as opposed to the mainly technological purposes of other research programs within computer science, robotics and synthetic biology. Moreover, they tend to attribute to this methodology the same heuristic features,<sup>1</sup> which, very schematically, can be listed as follows:

(a) The programmatic inversion of the established order between analysis of behaviour and construction of models — the SA directing researchers to first embed their basic hypothesis on life and cognition in working artificial systems, then examine the behaviours they produce.

(b) The theoretical hypothesis that distinguishes between organisation and physical-chemical realisation of living and cognitive systems, and claims that these systems and their phenomenology can be recreated by implementing the former in new physical media i.e. artificial “embodiments” and “embedments”.

(c) The emergentist framework which grounds living and cognitive behaviours not within the systems displaying them, but in the interplay between three basic organisational levels of life and cognition: the systems, their elemental components, and the environment(s) with which the systems interact.

(d) The correlated production of simple and generative models of living and cognitive systems, that is, models able to generate complex and unexpected behaviours through rather simple internal mechanisms. The latter models are designed to create complexity, not by themselves, but by participating in systems-components-environment(s) interactive dynamics.

Over the last few years considerable work has been done to extend the applicability of the SA within the domain of cognition (e.g. Pfeifer and Scheier 2000; Lungarella et al. 2003; Bedau 2003; Cañamero 2005; Dawson 2004; Froese and Ziemke 2009). These developments pertain to cognitive

<sup>1</sup> Cf. e.g. Luisi 2010, Pfeifer and Scheier 2000; Dawson 2002, 2004; Damiano and Cañamero 2010.

processes *lato sensu* (i.e. including affective and social processes)<sup>2</sup> and re-propose, in new terms, the epistemological question of how relevant this methodology is for the scientific study of natural complex behaviours.

At the origin of AL, this issue was mainly focused on living processes; as Langton put it, “the notion of studying biology via the study of patently non-biological things is an idea that is hard for the traditional biological community to accept” (Langton 1989, p. 52). Today this has significantly changed. Contemporary academic biological departments integrate areas of research such as bio-computation or chemical synthetic biology. This expresses the wide acceptance, within biology, of interdisciplinary collaboration grounded in SA. The SA is also being widely applied to the study of cognitive processes within communities such as behaviour-based robotics and embodied artificial intelligence and artificial life. The situation is still different for the study of affective processes, where embodied synthetic models are still a minority. The application of the SA to this area is, to a large extent, hindered by the lack of principled reflection regarding a number of unanswered epistemological questions, and this constitute an obstacle to the integration of the SA among the explorative practices accepted by the scientific community as sources of valuable insights for cognitive and behavioural sciences. These include, for example, questions such as: in what sense can systems endowed with artificial “embodiments” and “embedments” generate effective models of natural cognitive, affective, and/or social processes? In which ways and in what sense can the synthetic study of these processes provide significant advancements with respect to other models? Which are the criteria that permit to define the relevance of synthetic models for the inquiries of cognitive and behavioural sciences?

This article addresses these questions with the intent of taking a first step towards providing epistemological groundings to the application of the SA to the affective (and more generally other cognitive and behavioural) sciences. More concretely, our aim is twofold: (1) to define an epistemological framework (that is, a set of principles of knowledge) able to ground SA as a relevant methodology that also encompasses affective processes, in addition to other biological and cognitive processes already studied by embodied AI and artificial life; and (2) to derive from this framework a set of criteria of relevance for the synthetic modelling of all these processes, that is, criteria (i) valuable for all three main branches of AL (*soft*, *hard*, and *wet* AL) and (ii) able to define the relevance of their models for the scientific exploration of natural living affective and cognitive phenomena. This article pursues these objectives not through general and speculative dissertation, but by discussing a concrete case study: an interdisciplinary model of the development of attachment (and more generally affective) bonds in the area of developmental robotics.

Through the presentation of this case study, Section 1 introduces the epistemological issue we intend to face, as well as the epistemological approach we adopt. Section 2 describes in detail two principles of knowledge that we propose as an epistemological framework able to ground the application of the SA to affective processes. These principles are extracted from autopoietic biology founded by Humberto Maturana and Francisco Varela in the 70s (Maturana and Varela 1987). It is

worth noticing that our use of autopoiesis is different from the usual one. We do not refer to Maturana and Varela’s theory of life and cognition, as is often done in AL, to take inspiration for producing specific models of living and cognitive processes. Instead, we refer to Maturana and Varela’s *theory of scientific knowledge*, and draw on some of its elements. This is in order to provide the synthetic study of life and the synthetic study of cognition with a shared epistemological framework, able to offer them common criteria of relevance for the models they produce. In Section 3 we formulate and discuss the meaning of these criteria. In Section 4 we apply them to the developmental robotics model of attachment bonds, and discuss its contribution to developmental psychology. In section 5 we present the range of different forms of relevance that synthetic models can have with regard to biological, cognitive and behavioural sciences.

### 1. An interdisciplinary exploration of attachment bonds

Developmental robotics is a relatively recent area of research, located at the intersection of robotics and developmental sciences, within which the SA plays a crucial role. This area uses studies from developmental sciences not simply to construct “more autonomous, adaptable and sociable robotic systems” (Lungarella et al. 2003), but also to gain a deeper understanding of developmental processes. One of the programmatic goals of developmental robotics is to employ robots as tools to investigate, test and possibly further elaborate, in an interdisciplinary way, theories of development proposed by these sciences (Pfeifer and Scheier 1999, Pfeifer 2002, Sporns 2003, Cañamero 2005). This expresses the originality of this emerging area, which intends to use developmental theories not only for engineering purposes (to create better adaptive robotic systems), but also for genuinely scientific purposes, through the extension of the SA to cognitive developmental processes *lato sensu*.

Within this framework, our research on the development of attachment bonds provides a good example of research that intends to explicitly address these two goals (Cañamero et al. 2006), and this for the following main reasons.

Firstly, this kind of inquiry allows researchers to face one of the central issues that need to be successfully addressed to advance in the development of social robotics, that is, how to design robots that could learn from us, be accepted by us as social partners, and be able to adapt to our ever-changing social environments. The developmental approach deals with this issue on the basis of the idea that the most successful example of adaptation into our social and technological environment, without much prior knowledge, is given by infants. Following this approach, researchers have, for example, successfully managed to design robots that use algorithms to learn and adapt to new sensorimotor pairings (Berthouze and Lungarella 2004; Blanchard and Cañamero 2005; Andry et al. 2009; Hiolle et al. 2007). Other contributors have focused more closely on how developmental psychology describes infant development, investigating how infants explore and discover new features of the environment, particularly through drives like curiosity (Oudeyer et al. 2007) and seeking wellbeing through affect-driven interactions with objects and people (Blanchard and Cañamero 2005, 2006; Cañamero et al. 2006). Indeed, the latter contributions are addressing the issue of how positive affect, such as providing comfort, can promote an efficient and

<sup>2</sup> Cf. e.g. Nunez and Freeman 1999. In this article, when we refer to cognitive processes, we always refer to them *lato sensu*.

consistent learning experience, depending on the environment and especially the behaviour of the social partner.

Developmental robotics research on attachment bonds arose within this context, and, in accordance with the general orientation of the area, has a second goal: it also aims at contributing to the advancement of developmental psychology through the design of adaptive social robots modelled using scenarios, parameters and metrics that are also relevant to, and used by developmental comparative psychologists for, the study of attachment bonds in (human and non-human primate) infants. This modelling approach gives rise to robots that behave and interact with humans in ways that are comparable to young infants (in the specific variables of the phenomenon under investigation), and therefore could be used as tools to investigate and possibly further develop theoretical models about attachment bonds.

In the reminder of this section we briefly summarize the developmental robotics research on attachment bonds that we undertook within the Feelix Growing Project (<http://www.feelix-growing.org/>) in collaboration with developmental and comparative psychologists,<sup>3</sup> and are continuing within the ALIZ-E Project (<http://www.aliz-e.org/>). This work focuses on the mechanisms underlying the establishment and development of attachment bonds in the first two years of age, which has implications for all phases of affective development. As this paper is directed to introduce the epistemological issues related to the extension of SA to cognitive development, we do not provide here the technical details presented in other articles; for technical details, we refer the reader, for example, to (Cañamero *et al.* 2006; Hiolle *et al.* 2006, 2007; Hiolle and Cañamero 2007).

### 1.1 Development of attachment bonds

Human infants grow and discover their new environment most often accompanied by (or not far from) their mothers or primary caregivers. The skills they learn, and the objects and agents they encounter, are surely presented and assimilated within their cognitive and emotional experience with the constant help and assistance of these adult human beings alongside them. Attachment was originally defined (Bowlby 1969) as the affective tie between the infant and its primary caregiver which offers security and comfort when needed. In the last decades, developmental psychology has been trying to study how this affective tie influences cognitive and affective development of young children. This research has produced critical and revised versions of Bowlby's theory, which point out the complexity of attachment processes, as well as the dynamical and inter-individual character of the "dyad" child-caretaker (e.g. Tronick 2007; Keller 2008). They tend to describe this dyad as an inter-individual system whose components are involved in a dynamic co-determination which shapes the way the child interacts with his/her (social) environment, and re-shapes the way in which the care-giver(s) interact with the child. These critical developments strongly converge with the SA's assumption about the generation of complex behaviours,<sup>4</sup> and constitute the body of work that we took inspiration from. The remainder of

this section briefly introduces the properties of attachment bonds we used and the robot model we produced.

### 1.2 Attachment bonds in infants

One of the main roles of attachment bonds is to provide mechanisms that permit to regulate ("negative") affective state, and particularly arousal, setting the grounds for the development of emotion regulation later in life. We therefore focused on arousal and its regulation in our model. This essential variable was designed to relate to the notion of excitement as defined in (Sroufe 1996), which, in the early months of life, is neither a positive nor a negative emotion or affect, but refers to the level of internal activity and external stimulations experienced by the infant. A high and sustained level would be too demanding and challenging, while a low level would not give rise to fruitful behaviour. Thus, maintaining a good level of this variable is desirable. This internal variable is close to the concept of arousal (Berlyne 1960), relating to the theory of optimal arousal, and the inverted U-shape hypothesis (Anderson 1990), where mammals try to maintain on average their arousal at a middle level where their physiology is optimal. In our investigation of infant development, the notion of arousal is very appropriate, as it is used in developmental psychology to assess emotional intelligence in newborns and its development (Brazelton and Nugent 1995). However, the notion of arousal is often used as a dimension of the two or sometimes three dimensions usually adopted in models based on the circumflex model of emotions (Russel 1980), such as in (Breazeal and Scassellati 2002). In this kind of model, arousal is an orthogonal dimension to the valence of percepts and behaviours, and the model offers a one-to-one mapping from a two dimensional vector from the arousal/valence space to a predefined emotion. In our work, however, we do not use the notion of arousal in the same way as these models. Instead, we see arousal as a variable related to internal activity, in terms of learning experience, which is implicitly tied to external perceptions, some being more stimulating than others, according to familiarity and complexity.

The robotics model that we designed is based on the notion of arousal, which we associated with the learning experience of the robot and how stimulating or familiar the experienced environment is — namely, the current sensorimotor state. To this end, our model assesses whether the current percepts are being correctly memorised and recalled, and this directly influences the arousal level of the robot: novelty increases arousal, familiarity decreases it. The robot does not have explicit drives or motivations beyond exploring the environment, and its behaviour is a function of the level of arousal. The human, playing the role of a "caregiver", also has an impact on the arousal of the robot, in accordance with the secure base paradigm: the arousal level is decreased when the human provides comfort to the robot, either via direct tactile contact or by being present on its visual field. This robot-caregiver system is a dynamical system that present the essential elements needed to reflect and test the hypothesis concerning the attachment bonds and caregiving influences on it: unfamiliar events and stimuli increase the arousal and provoke distress, and the attachment figure can then relieve this distress with comfort. Whenever the arousal is low, the infant-

<sup>3</sup> Within the FEELIX GROWING project we worked in collaboration with Kim Bard and Jacqueline Nadel.

<sup>4</sup> Cf. the Introduction of this paper, point (d) and (Damiano and Cañamero 2009).

robot would keep exploring its environment as long as there are unknown features, in order to further its learning experience.

### 1.3 Robot Model

The robotic system is based on a few simple hypotheses, as stated above. Firstly, the robot's only "motivation" is to learn the features it can perceive in its environment. The level of arousal of the robot is calculated as a function of the familiarity and novelty of these features. The arousal rises when the robot is stimulated, and decreases when the attachment figure provides comfort visually or via tactile contact. When the level of arousal is low, the robot will seek stimulation and carry on exploring. The learning system of the robot uses two different well-known neural networks, a Kohonen Map (Kohonen 1997) and a Hopfield-like associative memory (Davey and Adams 2004, Hopfield 1982). The arousal level depends directly on the variation of the weights of the Kohonen Map, and on the accuracy of the associative memory. Indeed, a high variation of the weights is consequential of the robot discovering new features, and a mismatch between the output of the associative memory and the current perception is proportional to the novelty thereof. The arousal level is calculated as the exponential average of these two contributions over a predefined time window. When the caregiver touches the sensors on the back of the robot, the arousal level decreases exponentially and faster than it could increase whilst being over-stimulated. According to the arousal level and predefined thresholds, the robot behaves as follows. When the level is in a medium range, the robot remains still and attends to the current stimuli. Finally, when the level is high, due to too many and familiar stimuli, the robot is "distressed" and it will seek comfort from its caregiver.

Using this robot model, we undertook a series of studies focused on studying different aspects of the development of attachment bonds with one or more human caregivers.. These studies rest on the interdisciplinary design of experimental scenarios, with the aim to provide insight and feedback to all the different disciplines involved. Several crucial aspects of the development of attachment bonds were under inquiry. These included the development of different attachment profiles; the influence of these different profiles on exploratory behaviors; the role of attachment bonds in the development of sensorimotor associations; and the development of attachment bonds in the presence of multiple caregivers (Cañamero *et al.*, 2003; Hiolle *et al.* 2006, 2007; Hiolle and Cañamero, 2007).

### 1.4 The epistemological issue

The relevance of these studies for the construction of robots able to develop their skills and behaviours dependently on the interactions with their users is quite evident. However, considering the level of abstraction and simplification characterising the robotic architecture described above, the restricted possibilities of interaction of the "baby" robot with its human partner(s) and environment, the limited aspects of development of attachment bonds taken into account by the robot model, can we say that these studies are able to produce relevant feedback for human developmental psychology? Are there ways and conditions in which this "baby" robot can be fruitfully used to model and explore attachment phenomenology in humans?

If we refer to the widespread, and in our opinion incorrect, idea that a model system should "represent" the target system in all its aspects and behaviours, we have to answer in the negative. This notion, grounded in the classical "representationalist" scientific epistemology is also strongly questioned by the epistemological debate about synthetic modelling. The latter points out not only that necessarily scientific modelling fails in representing everything about target systems, but also that this is not its main goal. The basic purpose of scientific modelling is not to replicate the target system, but to investigate what are its relevant features. As in the case of synthetic modelling, the goal is to embed, in a model system, the scientific hypotheses about these features, and to test these hypotheses.

This argument has been proposed by both representationalist and non-representationalist epistemological approaches to AL. While the former use it to propose weak versions of the classical idea that scientific representation should be an exhaustive reflection of nature (e.g. Webb 2001), the latter use this argument to express the thesis that representationalism is not an appropriate epistemology for AL (e.g. Riegler 1992). According to the latter view, representationalist epistemological notions, based on the ideal of a science exploring objects independent from the observer, cannot orient the scientific practice grounded in the SA. Indeed, this methodology promotes a form of scientific knowledge which actively creates, and does not passively reflect, the phenomena explored. It refers to an observer who is the constructor, and not the old-fashion representationalist "spectator", of the systems he investigates. Moreover, the SA discards the representationalist dichotomy subjective/objective, as it proposes a way of doing science in which *facts* converge with *artefacts*, *discovery* corresponds to *invention*, *objective evidence* is not separable from *subjective construction*, and spontaneous manifestations of nature can be explored in the behaviour of artificial systems. As we showed in detail elsewhere (Damiano and Cañamero 2010), these remarks tend to lead non-representationalist approaches to AL that take inspiration from constructivist theories of scientific knowledge. These characterise science, in all its forms, as an activity of construction of objects of research (cf. e.g. Glasersfeld, 1995), and therefore propose to science epistemological notions and principles of knowledge which can be considered particularly appropriate for grounding, and supporting the scientific practice grounded in, the SA.

Our approach to the epistemological grounding of the SA belongs to this non-representationalist orientation. To address the issue of the relevance of synthetic models, like the robotic model presented above, for the inquiries of cognitive and behavioural sciences, we choose to refer to *autopoietic epistemology* (Maturana and Varela 1987). This is one of the best expressions of the constructivist epistemology developed across AL's scientific genealogy (Damiano and Cañamero 2010), and, as we argue, can provide shared groundings and common criteria of relevance to the synthetic study of life as well as to the synthetic study of affect and more generally cognition.

## 2. Grounding the SA in autopoietic biology

The connection between autopoietic cognitive biology and AL is strong. As already mentioned, Maturana and Varela provided more than an emergentist theory of life and an emergentist theory of cognition which are useful to inspire the production

of synthetic models of biological and cognitive processes. As often neglected (even by AL researchers who take inspiration from these theories), Maturana and Varela elaborated an explicit constructivist theory of scientific knowledge, which proposes the SA as the proper methodology to investigate both life and cognition, at the theoretical level as well as the experimental one. This autopoietic version of the SA, formulated more than a decade before the Langtonian one, is based on epistemological notions and principles which give expression to the intuition at the basis of Langton's AL program, and, in this sense, can be considered as appropriate epistemological groundings for AL's SA, both in the domain of life and in that of cognition.

On the basis of these considerations, we saw in Maturana and Varela's production a source of epistemological elements useful to provide a shared epistemological framework to the synthetic study of life and that of cognition. In particular, we extracted from autopoiesis two principles which, as we try to show in the remainder of this section, are particularly significant with regard to this goal. Below we summarize these (well-known, to a large part of the AL community) principles, to make the paper self-contained and help readers who might not be totally familiar with this approach.

### 2.1 Principle 1: Explaining = Constructing

The first principle proposes an operational definition of scientific explanation, according to which explaining a phenomenon amounts to proposing a mechanism able to produce it (cf. e.g. Maturana and Varela 1987, chapter 1). Visibly, the aim of this postulate is to extend the classic view of scientific explanation. It juxtaposes the traditional notion "explaining = predicting" to a constructivist one, which, proposing the equation "explaining = constructing", can be applied to systems exceeding scientific capabilities of calculation and prevision. Requiring models able not to *predict*, but to *generate* the natural processes under inquiry, the principle locates the focus of scientific explanation not on *actual*, but on *possible* behaviours of the systems explored. That is, it grounds a category of scientific descriptions which is particularly appropriate for living and cognitive systems, since the kind of characterization it proposes cannot be affected by these systems unpredictability. AL's SA can be legitimately included within this category, for it presents the basic distinctive features characterizing the paradigmatic constructive description of nature grounded by autopoiesis in this principle. In particular, it shares the distinctive features of the main example of constructive explanation provided by Maturana and Varela, namely, the autopoietic explanation of life. Very schematically: *raison d'être* (the natural phenomena it intends to describe are untreatable through the classical predictive modelling), epistemological grounding (the constructivist postulate according to which knowing scientifically means to build objects of research), heuristic gender (operational characterizations of the natural processes explored), procedure (definition of a generator for the phenomenology to be described, and exploration of the phenomenology it produces), and, finally, the appellation "synthetic".

In Maturana and Varela's literature, the introduction of this kind of scientific characterization is described as implying a long series of shifts in classical scientific epistemology, which produce a new emphasis not only on *construction* instead of

*representation*, on *generation* instead of *prediction*, on *possibility* instead of *actuality*, but also on *synthesis* instead of *analysis*. Indeed, in Maturana and Varela's production, the term *synthesis* defines the methodological orientation of autopoietic biology's theoretical program, just like, in Langton's literature, it defines the methodological orientation of AL's program.

On the basis of its principle of scientific explanation, autopoietic biology plans to formulate a procedurally new definition of life, which, instead of listing the main features of living systems, specifies a dynamical mechanism able to produce their phenomenology. Maturana and Varela call this kind of definition "synthetic", to distinguish it from the traditional "analytic" definitions of life presenting detailed lists of properties. The condition that this synthetic definition has to satisfy to be considered an appropriate explanation of life is expressed in terms of its theoretical productivity. The mechanism that it specifies has to show the capability of creating, from a set of elemental components, an entire biological domain. That is, it has to manifest the ability of generating, by the dynamical coordination of a set of elements, a minimal cellular system with its characteristic phenomenology. That is: not only cellular self-production, but, through this, also reproduction and evolution, to the extent to be able of producing, step by step, a differentiated living domain, as complex and populated as the terrestrial one.

This is the kind of scientific modelling of the living that Langton's characterization of AL intends to implement too, not at the level of a purely theoretical construction, but at that of an empirical one: the synthesis of "any and all biological phenomena, from viral self-assembly to the evolution of the entire biosphere", without restriction to carbon-chain chemistry. As in the case of Maturana and Varela, Langton's program is that of a constructive and universal biology, which converges with autopoietic biology not only on the basic epistemological principle of scientific explanation, but also on the principle of biology's universalisation. "Life is (...) a result of organization of matter, rather than something that inheres in the matter itself" (Langton 1989, p. 53).

### 2.2 Principle 2: Organization ≠ Structure

The second autopoietic principle we consider pertinent for the epistemological grounding of the SA is a theoretical postulate with a significant epistemological value. Its basic content is the distinction between two notions — *organization* and *structure*. Simplifying the original autopoietic formulation (Maturana and Varela 1987, chapter 2), we can put it as follows: the *organization* of a living system is its relational frame, that is, the network of relations which define the system as a unity of components; the *structure* of a living system is its materialization, given by the actual components and their interconnections.

This distinction is not a theoretical novelty introduced by Maturana and Varela. A first complete formulation can be attributed to Jean Piaget (1967, chapter 4), who proposed this conceptual distinction as the theoretical key to comprehend biological systems as dynamical, since it corresponds to the distinction between the invariant and the variant aspects of their dynamics. Piaget remarked that living systems can be considered dynamical systems endowed with a peculiarity: all their elementary components permanently change, while systems, as relational unities of components, remain. This, as

Piaget pointed out, can be affirmed at both the ontogenetic and the phylogenetic levels. The relational unity remains unchanged not only in the permanent flux of physical-chemical components typical of biological organisms, but also during the ontogenetic transformations which can make a living system unrecognisable from one observation to the next. Moreover, this relational unity is transmitted through reproduction and remains unchanged generation after generation. Indeed, this relational unity is the invariant of the biological dynamics and therefore the lowest common denominator of living systems. Distinguishing this invariant relational frame from the changeable materializations of living systems, and determining its configuration, amounts to isolating an element which can be used to define the class of biological systems.

These remarks point out the epistemological relevance of the distinction between organization and structure, which is at least two-fold. Firstly, this distinction allows biological research to hypothesize a defined mechanism for living dynamics (i.e. a mechanism creating organizational invariance through permanent structural variation), and therefore opens the possibility of a constructive explanation of life. Secondly, it generates significant insights about the SA's relevance to the study of natural living and cognitive processes, as it implies that: (a) in principle the materialization (structure) of living systems can be manifold; (b) artificial systems displaying the same organisation as living systems, and realising it in a different structure, have to be considered as belonging to the class of living systems.<sup>5</sup>

Thus, the autopoietic distinction between organisation and structure offers a theoretical ground to the thesis — “the big claim” — through which Langton expresses AL's aspiration: “a properly organized set of artificial primitives carrying out the same functional roles as the bio-molecules in natural living systems will support a process that will be ‘alive’ in the same way that natural organisms are alive. AL will therefore be genuine life — it will simply be made of different stuff than the life that has evolved here on Earth.” (Langton 1989, p. 69)

### 2.3 Autopoiesis and the extension of the SA to the domain of cognition

Autopoietic biology does not limit itself to formulating principles that support Langton's initial AL program. It also supports the extension of this program to cognitive processes. The intent of “naturalising cognition” led Maturana and Varela to identify living systems as cognitive systems, since their general process of self-production (i.e. autopoiesis) corresponds to a permanent process of interaction with the environment and other systems (structural coupling) that allows living systems to survive. The conceptualisation of this process as a process of cognition is at the basis of the cognitive biology that Maturana and Varela developed as an extension of their theory of life, fathering the nascent “embodied cognitive science” (Clark 1999). According to this view, the phenomenology that has to be produced by the autopoietic synthetic definition of life includes not only all the biological, but also all the cognitive

<sup>5</sup> The thesis of the multiple material realization of organization implies a convergence between autopoiesis and functionalism. There is no room here for a detailed comparison. However it is worth noticing that autopoiesis and functionalism have different views about the implications of this thesis. For example, autopoiesis, differently from functionalism, emphasizes the dependence of cognition on the agent's embodiment.

phenomenology *lato sensu* (Maturana and Varela 1987). In this sense, the autopoietic principle of the constructive explanation and the autopoietic distinction between organisation and structure offer a grounding framework not only to the synthetic study of life, but also to the synthetic study of cognition *lato sensu*.

### 3. Two criteria of relevance for the SA

The two autopoietic principles presented above can be transformed into two criteria for use in determining the relevance of the SA's implementations to the study of life and cognition.

#### 3.1 – “Explaining = Constructing”: phenomenological relevance<sup>6</sup>

From the principle of scientific explanation extracted from autopoietic biology's production (P1: *To explain scientifically is to provide a mechanism able to produce the phenomenology to be explained*) can be derived a criterion of “phenomenological relevance” for synthetic models of natural living and cognitive phenomena, according to which:

(C1) *A synthetic model is relevant at a phenomenological level if it provides a mechanism which produces (according to explicit parameters) the same phenomenology as the living or cognitive phenomenology under inquiry.*

The appellation “phenomenological relevance” expresses the fact that this criterion requires only a relation of identity (defined by explicit parameters) between the phenomenology produced synthetically and the natural phenomenology under inquiry. This means that (C1) does not impose any constraints on the biological plausibility of the synthetic mechanism by which the phenomenology under exploration is produced. Therefore, if (C1) is not correlated to a criterion which requires the biological plausibility of synthetic models, and specifies what this plausibility consists of, then (C1) cannot warrant that these models offer a biologically plausible explanation of the target processes, and that they do not simply imitate the phenomenology under inquiry.

However, from autopoietic biology we can also extract a principle to differentiate phenomenologically relevant models on the basis of their respective operational explanatory powers — that is, on the basis of their capability of providing an operational explanation of the phenomena under inquiry. This principle, belonging to the autopoietic theory of scientific explanation (Maturana and Varela 1987, chapter1; Maturana 1988), associates the operational explanatory power of a model to its “progressive” character, that is, its capability of producing, besides the phenomenology under inquiry, also other phenomena belonging to the same domain.<sup>7</sup> In this sense, autopoietic epistemology provides a principle of evolution to the phenomenologically relevant synthetic modeling of living and cognitive processes, which can orient the choice between different models referred to the same phenomenological domain. That is, on the basis of the quantity of supplementary phenomena, that they are able to produce, these models can be considered more or less progressive (that is, more or less operationally explanatory) than another models.

<sup>6</sup> Here the adjective ‘phenomenological’ has the meaning of ‘relative to the phenomenology under inquiry’.

<sup>7</sup> We use the term ‘progressive’ in accordance with Lakatosian philosophy.

According to this principle, we have to distinguish two basic kinds of phenomenologically relevant models. They can be respectively defined as follows:

(1) *minimal phenomenological models*, which produce only the phenomenology under inquiry, and therefore have a minimal operational explanatory power;

(2) *progressive phenomenological models*, which produce, besides the phenomenology under inquiry, other phenomena belonging to the same domain, and have an operational explanatory power proportioned to the quantity of supplementary phenomena produced.

Evolution towards better phenomenologically relevant models corresponds to evolution towards models endowed with a higher operational explanatory power, but not necessarily towards biologically plausible models. Indeed, even if a greater operational explanatory power could be considered as a clue of greater biological plausibility, this last remains uncertain in absence of a criterion which specifies what this plausibility consists of. A synthetic progressive model, in itself, could be useful for the traditional scientific exploration of living and cognitive processes as it could offer not a biologically plausible explanation of these processes, but a source of inspiration for the production of hypotheses about the mechanisms underlying them.

### 3.2 “Organization ≠ Structure”: relevance in the strong sense

As pointed out before, the autopoietic distinction between organization and structure implies that (P2.i) *All living and cognitive systems share the same organisation, but not necessarily the same structure*, and therefore that (P2.ii) *Artificial systems which display a different structure, but the same organisation as living and cognitive systems, have to be considered legitimately belonging to the class of living systems*. Thus, the autopoietic distinction between organization and structure produces a criterion of relevance for synthetic models of living and cognitive systems. In fact, in accordance with (P2.ii), the former can be considered *strong models* of the latter if they share the same organisation, since, in this case, they constitute *specimens* of the class of living and cognitive systems.

We can refer to this criterion as to a criterion of *organisational relevance*, which warrants the biological plausibility of synthetic models. Associated to the criterion of *phenomenological relevance*, it produces the criterion of “relevance in the strong sense”:

(C2) *Synthetic models are relevant in the strong sense if, besides providing mechanisms which generate the phenomenology under inquiry (phenomenological relevance), they present (according to some explicit theory of living and/or cognitive organisation) the same organisation as living and cognitive systems*.

Satisfying this criterion is indeed a hard challenge for AL, which, of course, always has to be faced referencing one or more theories of biological and/or cognitive organisation, and always in an approximate way due to the intrinsic limits of these theories, the varieties of their interpretations, and the limited possibilities of their implementation. In this sense, relevance in the proper sense has to be considered for artificial life more a regulative ideal than a concretely attainable goal.

## 4. Interactive phenomenological relevance

If attachment phenomenology is defined as the closed set of phenomena normally used to exemplify it (e.g. seeking the proximity of the caregiver, developing stress in situation of separation and developing different attachment profiles depending on caregiver behaviour), then the robot model can be considered to (roughly) satisfy (C1). But, as far as we tested it, this model does not have a progressive character, and cannot be considered biologically plausible according to (C2). Therefore we are led to attribute it a *minimal phenomenological relevance* with regard to attachment behaviours, and to consider it as simply imitating them.

However the robot model does more than this when the system under consideration is the *human-robot interacting dyad*. Using this model in experiments involving humans in the role of caregivers, the resulting evidence suggests that it has further scientific potential, related not to its operational explanatory power or its biological plausibility, but to its capability of dynamically interacting with human agents. In fact, the “baby” robot appears able to engage humans in *interactive dynamics* which can be of scientific interest for the developmental psychology inquiry on attachment bonds (Hiolle *et al.* 2008). That is, it offers to developmental psychology the possibility of experimentally manipulating and exploring, in human agents, aspects of the attachment phenomenology that can be difficultly accessible in the classical psychological scenarios of research. An example of these processes can be found in human caregivers’ reactions to different attachment profiles. This is an aspect of the attachment phenomenology that developmental psychology could study through robot models like the one presented above, as emerged from our interdisciplinary exploration of attachment bonds.

These remarks lead us to introduce a new kind of minimal phenomenological model. These can be defined as *interactive phenomenological models*: models able to synthetically produce the phenomenology under inquiry, and, through the expression of this phenomenology, to engage natural biological and/or cognitive systems in interactive dynamics which (according to some explicit parameter) prove interesting for the scientific exploration of the natural phenomenology under inquiry.<sup>8</sup> As such, this kind of minimal phenomenological model can concretely contribute to biological, behavioural and cognitive science’s inquiries on natural living and/or cognitive processes, as synthetic tools that, through their capability of interacting with natural living and/or cognitive systems, can support the experimental manipulation and investigation of their processes.

## 5. Conclusions

This article proposes a constructivist solution to the issue of providing epistemological groundings for the application of the SA to affective and social processes. This solution consists of an epistemological framework extracted by autopoietic epistemology, able to provide to both the synthetic study of life and the synthetic study of cognition with (1) shared grounding principles of knowledge, and (2) shared general criteria useful to define the relevance of (soft, hard, and wetware) synthetic

<sup>8</sup> Soft AL’s production is rich of examples of virtual agents interacting with human agents, and able to engage them in interactive dynamics which could be interesting from a scientific point of view. Examples of synthetic systems able to interact with natural systems are emerging also in wet AL (cfr. e.g. Kaneda *et al.* 2009 about interactions between synthetic models of minimal cells and cultured cells).



models for the exploration of life and cognition. We showed how these criteria open a space of relevance for the synthetic modelling of life and cognition defined by two extremes. The “lower” extreme is *minimal phenomenological relevance*, which characterises models that, by reproducing synthetically the natural phenomenology under inquiry, offer an operational explanation of these process, but, as they do not have biological plausibility, have to be considered synthetic imitations of them. The “upper” extreme, which has to be considered more a regulative ideal, than a concretely attainable goal, is (2) *relevance in the strong sense*. It characterises models that reproduce synthetically the natural phenomenology under inquiry, and, as they display the same organisation as living and/or cognitive systems, can be considered to belong to the class of living and/or cognitive systems. We argued that, within this space, AL can produce two kinds of synthetic models that could be of interest for its interdisciplinary cooperation with biological, behavioural and cognitive science. The first is given by models characterised by a *progressive phenomenological relevance*, that is, the capability of producing not only the phenomenology under inquiry, but also other phenomena belonging to the same domain. These models have a significant operational explanatory power, and, dependent on their biological plausibility, can prove useful for biological, behavioural or cognitive sciences as a source of inspiration for the definition of the mechanism underlying the phenomenology under inquiry. The second kind is given by models characterised by a *interactive phenomenological relevance*, that is, the capability of producing synthetically the phenomena under inquiry, and, through this, engaging natural systems in interactive dynamics that prove useful to experimentally investigate, in natural systems, at least some aspects of the phenomenology under inquiry.

## Acknowledgments

This work has received support from the EC-funded Projects FEELIX GROWING (FP6-IST-i) and ALIZ-E (FP7-ICT-248116).

## REFERENCES

- Anderson, K. J. (1990). Arousal and the inverted-U hypothesis. *Psychological Bulletin* 107, 1:96 – 100.
- Andry, P., Garnault, N. and Gaussier, P. (2009). Using the interaction rhythm to build an internal reinforcement signal. In *Proceedings of the Ninth Int. Conf. on Epigenetic Robotics*. Lund University Cognitive Studies.
- Bedau, M. (2003). Artificial life: organization, adaptation and complexity from the bottom up. *Trends in cognitive science*, 7:505-512.
- Berlyne, D. E. (1960). *Conflict, Arousal and Curiosity*. Mc Graw-Hill Book.
- Berthouze, L. and Lungarella, M. (2004). Motor skill acquisition under environmental perturbations. *Adaptive Behavior* 12: 47–63.
- Blanchard, A. and Cañamero, L. (2005). Using visual velocity detection to achieve synchronization in imitation. In *Third Intl. Symposium on Imitation in Animals and Artifacts*, SSAISB Press, 26–29.
- Bowlby, J. (1969). *Attachment and loss. Vol 1: Attachment*. Basic Books.
- Breazeal, C. and Scassellati, B. (2002). Challenges in building robots that imitate people. In: Dautenhahn, K. and Nehaniv, C. editors, *Imitation in Animals and Artifacts*. MIT Press, Cambridge, MA. 14, 363–390.
- Brazelton, T. B. and Nugent, J. (1995). *Neonatal Behavioral Assessment Scale*. Cambridge University Press.
- Cañamero, L. (2005). Emotion understanding from the perspective of autonomous robots research. *Neural networks*, 18:445-455.
- Cañamero L. (2002). Playing the emotion game with Felix. What can a Lego robot tell us about emotion? In: Dautenhahn, K., Bond, A.H., Cañamero, L., Edmonds, B., *Socially Intelligent Agents*. Kluwer
- Cañamero, L. and R. Aylett R. editors (2008). *Expressive characters for social interaction*. John Benjamins, London.
- Cañamero, L., Blanchard A., and Nadel J. (2006). Attachment bonds for human-like robots. *International Journal of Humanoid Robotics*, 3:301-320.
- Clark, A. (1999). An embodied cognitive science? *Trends in Cognitive Science*, 3, 9:345-351.
- Davey, N. and Adams R. (2004). High capacity associative memories and connection constraints. *Connection Science* 16, 1, 47–65.
- Damiano, L. and Cañamero L. (2010). Constructing Emotions. *Proceedings of AIB 2010*.
- Dawson, M. R. W. (2002). From embodied cognitive science to synthetic psychology. In: *Proceedings of the First IEEE International*.
- Dawson, M. R. W. (2004). *Minds and machines*. Blackwell, London.
- Froese, T and Ziemke T. (2009). Enactive artificial intelligence. *Artificial Intelligence*, 173: 466–500.
- Glaserfeld, von, E. (1995). *Radical constructivism*. The Falmer Press. London.
- Hiolle, A., Bard, K. and Cañamero, L. (2006). Assessing human responses to different robot attachment profiles. *Proceedings of the 18th International Symposium RHIC*, pp. 251-257.
- Hiolle, A. and Cañamero L. (2008). Why should you care? An arousal-based model of exploratory behavior for autonomous robots. In *Proceedings of the 11th International Conference on Artificial Life*, 242-248, MIT.
- Hiolle, A. and Cañamero, L. (2007). Developing sensorimotor associations through attachment bonds. In *Proceedings of the 7th International Workshop on Epigenetic Robotics*, 45–52.
- Hiolle, A., Cañamero, L. and Blanchard, A. (2007). Learning to interact with the caretaker: a developmental approach. In *Proceedings of the 2nd ICACII*, 425–436.
- Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America* 79, 8, 2554.
- Kohonen, T. (1997). *Self-Organizing Maps*. Springer-Verlag.
- Kaneda, M. et al. (2009). Direct formation of proteo-liposomes by in vitro synthesis and cellular cytosolic delivery with connexin-expressing liposomes. *Biomaterials*, 30: 3971–3977.
- Keller, H. (2008). Attachment-Past and present. But what about the future? *Integr Psych Beav* 42: 406-415.
- Langton, C. (1989). Artificial life. In: Boden M editor (1996), *The Philosophy of Artificial Life*. Oxford University Press, 39-94.
- Luisi, P. L. (2010). *The synthetic approach in biology*. Manuscript.
- Lungarella, M., Metta, G., Pfeifer, R. and Sandini, G. (2004). Developmental robotics: a survey. *Connection Science*, 1–40.
- Maturana, H. and Varela, F. (1987). *The Three of Knowledge*. Shimbhala.
- Nunez, R. and Freeman, W. J. editors (1999). *Reclaiming cognition*. Imprint.
- Oudeyer, P.-Y., Kaplan, F. and Hafner, V. (2007). Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation* 2, 11.
- Piaget, J. (1967). *Biologie et connaissance*. Gallimard, Paris.
- Pfeifer, R., Lungarella, M. and Sporns, O. (2008). The synthetic approach to embodied cognition. In Calvo P. and Gomila T. editors. *Handbook of cognitive science. An embodied approach*. Elsevier.
- Pfeifer, R. and Scheier, C. (2000). *Understanding intelligence*. MIT Press, Cambridge, MA.
- Riegler, A. (1992). Constructivist Artificial Life, and Beyond. In: McMullin, B. editor. *Proceedings of the Workshop on Autopoiesis and Perception, Dublin City University: Dublin*, pp. 121–136.
- Russel, J. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology* 39: 1161–1178.
- Sroufe, L. A. (1996). *Emotional Development: The Organization of Emotional Life in the Early Years*. Cambridge University Press.
- Tronick, E. (2007). *The neurobehavioural and socio-emotional development of infants and children*. Northon & Company.
- Webb, B. (2001). Can robots make good models of biological behaviour? *Behavioral and Brain Sciences*, 24: 1033-1050.

# Application of Small-World Mutation Topologies to an Artificial Life System

Alastair Droop<sup>1,2</sup> and Simon Hickinbotham<sup>1,3</sup>

1: YCCSA, Ron Cooke Hub

2: YCR Cancer Research Unit, Department of Biology

3: Department of Computer Science

The University of York, Heslington, York YO10 5DD, UK

email: alastair.droop@york.ac.uk & sjh@cs.york.ac.uk

## Abstract

The mutation networks observed in biological systems have the properties of small-world networks. These properties of short average path length and high transitivity confer a favourable exploration of mutation space. Any evolvable string-based ALife system (for example stringmol, typogenetics, Tierra, or Avida) uses a substitution network either implicitly or explicitly. Current ALife simulations use either regular or random mutations schemes. We have previously discussed the requirement for small-world substitution networks for ALife simulations. In this paper, we explore the effects of rewiring the stringmol mutation lattice on the evolution of a self-replicating molecule.

## Introduction

Mutation is an essential component of any evolvable system, allowing it to explore its fitness landscape and therefore to evolve. The evolutionary dynamics of a system are thus critically dependent upon its mutation strategy.

Amino acid substitution matrices (Dayhoff et al., 1978; Henikoff and Henikoff, 1992) give an indication of the likelihood of observing an amino acid substitution in homologous proteins. Ideally, a substitution matrix should allow any token to mutate to any other token relatively easily (thus allowing a rapid exploration of the fitness landscape); whilst simultaneously favouring mutations to tokens of similar function (thus minimising the chance of deleterious mutations). Networks that exhibit these properties of short average path length and high clustering coefficient were described by Watts and Strogatz (Watts and Strogatz, 1998). We have previously demonstrated that biological mutation networks exhibit these small-world properties (Droop and Hickinbotham, 2011).

Although biological mutation networks exhibit small-world properties, ALife simulation mutation schemes do not. The typogenetics (Gwak and Wee, 2007) and Avida (Johnson and Wilke, 2004) systems use essentially random mutation schemes. By contrast, the stringmol (Hickinbotham et al., 2010a,b) and Tierra (Ray, 1991) systems use regular mutation networks. Figure 1 shows the mutation lattices used by the stringmol and Tierra systems.

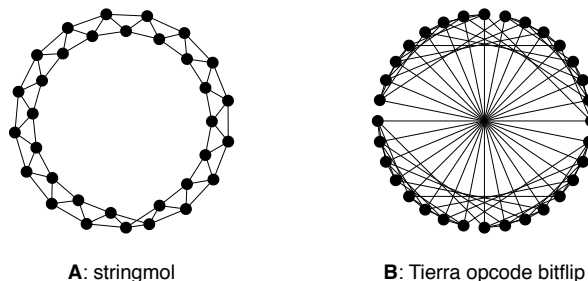


Figure 1: The mutation networks used by the stringmol and Tierra ALife simulations. The stringmol lattice shown here has  $k = 4$ .

The stringmol mutation network is constructed as a complete lattice with each node connected to its  $k$  nearest neighbours (in this case  $k = 4$ ). The Tierra network is based upon binary bit flip operations: each opcode can flip a single binary digit. Any mutation scheme can be represented by a graph where possible substitutions between tokens are represented as edges. The Tierra opcode lattice in figure 1 shows that although the neighbours for each opcode were carefully chosen to allow ‘sensible’ mutations, the mutation network topology is nonetheless regular.

To test this idea, we implemented a small-world mutation network topology for the stringmol system using the rewiring scheme devised by Watts and Strogatz (Watts and Strogatz, 1998). Here, we present the findings from this study, and discuss the effects of different mutation models upon the stringmol system.

## Methods

Multiple networks were created for the stringmol system using the Watts and Strogatz (1998) rewiring model as described previously (Droop and Hickinbotham, 2011). The stringmol regular mutation lattice with  $k = 4$  was used to perform all analyses. The mutation rate is fixed for all trials. Four different rewiring probabilities ( $p_{\text{reg}} = 0$ ,  $p_{\text{small}} = 0.1$ ,  $p_{\text{mid}} = 0.3$  and  $p_{\text{rand}} = 1$ ) were used when cre-

ating networks. The  $p_{\text{reg}}$  network is the same regular lattice as used previously in stringmol. The  $p_{\text{small}}$  and  $p_{\text{mid}}$  networks fall within in the small-world region for networks of this size (Droop and Hickinbotham, 2011). The  $p_{\text{rand}}$  network is completely random. 100 rewiring replicates were created for  $p_{\text{small}}$ ,  $p_{\text{mid}}$  and  $p_{\text{rand}}$ ; whilst only a single trial of  $p_{\text{reg}}$  was performed (as there is only one possible network). For each replicate experiment, the stringmol simulation run for a maximum of  $1.2 \times 10^9$  time steps with 300 trials.

Summary statistics were collected for each trial. Four statistics were used to describe each individual trial. The *life time* is the total number of time steps over which the trial survives. The *epoch count* is the number of epochs (defined as a continuous period of time in which a particular string species is the most common). The *epoch length* is the length of each epoch. The *edit distance* is a measure of the mutational distance that the trial has been able to cover. The edit distance is calculated as the Smith-Waterman alignment score between the initial string and the dominant string (only counting strings greater than 10 letters, thus ignoring short ‘pathogenic’ strings) for the last epoch of length  $\geq 50000$ , normalised to the length of the string.

## Results

Box plots (McGill et al., 1978) of the results for the summary statistics outlined above are given in figure 2. The trend across figures 2A, B and C is consistent: the more random the mutation network, the shorter the simulation life time and fewer (shorter) epochs are present. The reduction in life time indicates that large amounts of energy are wasted in the simulation (presumably on harmful mutations). Similarly, the reduction in the epoch count indicates that fewer successful species are created during the trial. Figure 2D, however, shows that the mutational distance covered by the runs *increases* with increasing randomness in the mutation networks: this demonstrates that although shorter, the runs with more random mutation schemes can produce more varied successful species.

## Summary & Conclusion

Taken together, the results shown here suggest that the optimal mutation strategy for the stringmol system is neither at  $p_{\text{reg}}$  or  $p_{\text{rand}}$ ; rather somewhere in between: in the small-world rewiring region. This work provides an experimental validation of the argument presented previously (Droop and Hickinbotham, 2011).

## Acknowledgements

Alastair Droop is funded by Yorkshire Cancer Research. Simon Hickinbotham is funded by the Plazmid project, EP-SRC grant EP/F031033/1.

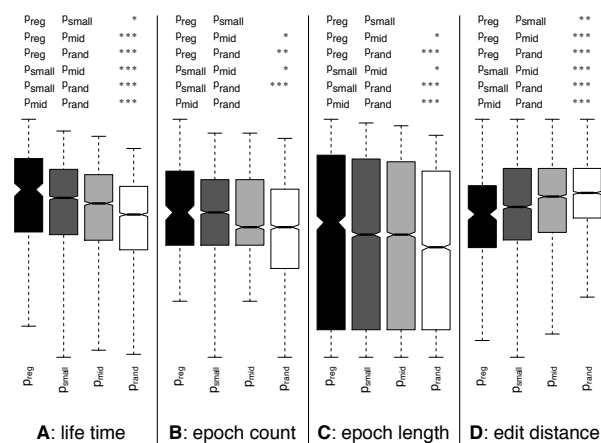


Figure 2: Box plots of four summary statistics for each experiment. Each plot is drawn using a logarithmic  $y$ -axis (values omitted for clarity). 1, 2 and 3 stars represent t-test  $p$ -values of  $\leq 0.05$ ,  $0.005$  and  $0.0005$  respectively.

## References

- Dayhoff, M. O., Schwartz, R., and Orcutt, B. C. (1978). A model of evolutionary change in proteins. *Atlas of protein sequence and structure*, 5(supplement 3):345–358.
- Droop, A. and Hickinbotham, S. (2011). Properties of biological mutation networks and their implications for artificial life. *ALife Journal*, (in press).
- Gwak, C. and Wee, K. (2007). Construction of hypercycles in typogenetics with evolutionary algorithms. In *Proceedings of the 9th European conference on Advances in artificial life*, ECAL’07, pages 1060–1068, Berlin, Heidelberg. Springer-Verlag.
- Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, 89(22):10915–10919.
- Hickinbotham, S., Clark, E., Stepney, S., Clarke, T., Nellis, A., Pay, M., and Young, P. (2010a). Diversity from a monoculture: Effects of mutation-on-copy in a string-based artificial chemistry. In *ALife XII*, pages 24–31. MIT Press.
- Hickinbotham, S., Clark, E., Stepney, S., Clarke, T., Nellis, A., Pay, M., and Young, P. (2010b). Specification of the stringmol chemical programming language version 0.2. Technical Report YCS-2010-458, The University of York.
- Johnson, T. J. and Wilke, C. O. (2004). Evolution of resource competition between mutually dependent digital organisms. *Artificial Life*, 10(2):145–156.
- McGill, R., Tukey, J. W., and Larsen, W. A. (1978). Variations of box plots. *The American Statistician*, 32(1):12–16.
- Ray, T. S. (1991). An approach to the synthesis of life. In *Artificial Life II*, pages 371–408. Addison-Wesley.
- Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684):440–442.

# Quantifying Normative Behaviour and Precariousness in Adaptive Agency

Matthew D. Egbert<sup>1\*</sup>, Xabier E. Barandiaran<sup>2</sup>

<sup>1</sup> Center for Computational Neuroscience and Robotics, University of Sussex, Brighton, U.K.

<sup>2</sup> Centre de Recherche en Epistemologie Appliquée, CNRS / Polytechnique, Paris, France.  
mde@matthewegbert.com

## Abstract

An essential feature of autonomous adaptive agency is that a system behaves according to an intrinsic norm. In this paper, we illustrate and clarify this notion of “behavior according to an intrinsic norm” with a minimalistic model of agency. We present a minimal metabolic system whose auto-catalytic dynamics define a viability region for different concentrations of available resource or ‘food’ molecules. We initially consider the availability of food as a control parameter for metabolic dynamics. A bifurcation diagram shows that for fixed values of available food, there exists a *viability region*. This region has a non-zero stable equilibrium and a lower boundary that takes the form of an unstable equilibrium—below which, the tendency of the system is towards “death”, a stable equilibrium with a zero concentration of metabolites. We define the viability region as that in which the system tends toward the “living” stable-equilibrium. Outside of this region, in the *precarious region*, the system may live for some time but will eventually die if the food concentration does not change. With a precise definition of system-determined death, living, precarious and viable regions we move on to reconsider the available concentration of resources ( $[F]$ ), not as a free parameter of the system but as modulated by organismic behaviour. By coupling the metabolism to a behavioural mechanism, we simulate a stochastic, up-resource gradient climbing behaviour. As a result, the effect of behaviour on the viability space can be mapped and quantified. This lets us move closer to defining adaptive action more precisely as that course of behaviour whose effect is in accordance with an intrinsic normative field.

## Introduction

The way in which living systems (from bacteria to humans) actively regulate their relationship with their environments strongly contrasts with inanimate objects. This *agency* is widespread in nature and it continues to capture the attention of philosophers, theoretical biologists, psychologists and roboticists alike, for it has proven to be a difficult property to define, model or synthesise.

The notion of agency often carries with it closely related and traditionally problematic notions such as normativity, adaptivity, individuality, teleology, intentionality, goal-directedness or free-will. Artificial life modelling techniques are well suited to provide a bottom-up approach ca-

pable of conceptually clarifying the systemic character of the properties associated with agency, its origins and nature.

After reviewing a wide variety of definitions and uses of the term ‘agency’ ranging from biology to robotics, Barandiaran et al. (2009) define agency as follows:

“an agent is an autonomous organization capable of adaptively regulating its coupling with the environment according to the norms established by its own viability conditions.” (p.376)

In this paper we attempt to make more explicit what is meant by the expression “according to the norms established by its own viability conditions”. Similar expressions have been used by Di Paolo (2005); Barandiaran and Moreno (2008); Skewes and Hooker (2009) but no model has yet been developed to illustrate and describe in detail the meaning of this expression (and others closely associated with it). The goal of this paper is to make progress in this direction using a minimalist model that can help understand and scientifically articulate a formal and quantitative definition of agency. To this end, we present a model that exemplifies the key concepts of “normative behaviour” in the context of agency. To further contextualize the model and its interpretation, in the next section we introduce the conceptual (i. e. philosophical) and theoretical problem and two contemporary approaches to it. We then introduce the design specifications of the model and analyze its dynamics and their interpretation in terms of normativity, precariousness, adaptivity and viability.

## Autonomous agency and normativity: some dynamic requirements

The issue of natural agency and norms is attracting increasing attention (Frankfurt, 1978; Burge, 2009; Di Paolo, 2005; Skewes and Hooker, 2009; Barandiaran et al., 2009; Silberstein and Chemero, 2011) and Artificial Life is very well suited to make some conceptual progress on key aspects of agency and its origins. In fact, minimal models of agency have been a recurring topic in the field (from protocellular models to robotics).

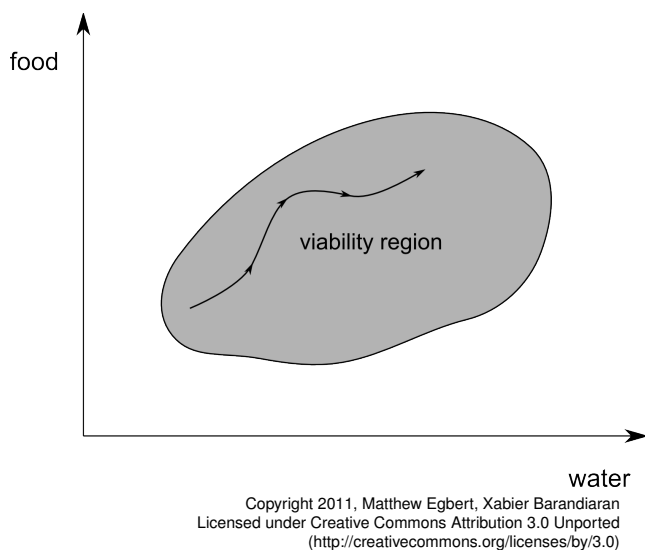


Figure 1: Classical picture of a Viability region defined for two essential variables (food and water), outside the region the system will die. Viable trajectories are those that remain within the boundaries of the viability region. Model generally focus on designing a control system that generates the appropriate trajectories inside the viability region but the region itself is given (e. g. arbitrarily designed or experimentally determined).

Arguably, minimal forms of agency (like chemotaxis) encapsulate some of the most important properties of “higher” levels of agency (such as human agency). One such property is *normativity*: i.e. the dimension of behaviour in which *value* comes into play—in which actions are good or bad, adaptive or maladaptive, appropriate or inappropriate (Christensen and Bickhard, 2002; Barandiaran and Moreno, 2008; Burge, 2009). While artificial systems can be judged to operate in relation to norms, these norms have (thus far) always been defined by the designer of the artificial system or interpreted by an external observer or user. In other words, what is good or bad functioning for a robot, a car or a coffee machine has been a matter of the design specifications which are largely independent of the structure and organization of the artifact. This is unlike biological organisms that respond to norms that are more closely related to the organization of the organism itself and what is (or is not) conducive to its ongoing operation.

Philosophers and scientists have tried to justify this normative dimension of natural agency in two ways. The most popular is the evolutionary (Millikan, 1989) approach in which a behaviour is considered to be normative or adaptive if it has been selected by evolution. In this view adaptation is ultimately a result of natural selection and it is only as a result of a process of selection that a character or process (e. g. a pattern of behaviour) can be said to be adaptive or maladaptive. This evolutionary approach to etiology, faces nu-

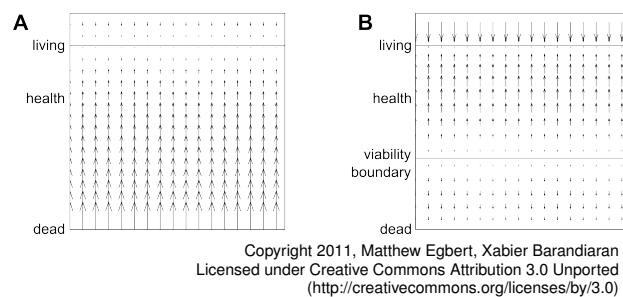


Figure 2: The viability boundary is an unstable equilibrium between living and dead states.

merous problems. One of them is how to categorise the first instance of a particular adaptive (i.e. norm following) behaviour? If a norm depends on an evolutionary selective history, then the first case of a “norm following” behaviour does not qualify as norm-following until it has been selected. This is clearly unsatisfactory. A criteria that is independent from history and is instead grounded on the very organization of the system and its ongoing dynamics seems better suited—indeed required—if we are to derive a consistent definition of adaptivity and normative behavior. This is precisely the motivation underlying the main alternative approach to normativity and adaptation. The *organizational approach* (as it might be called) puts at its center the idea of autonomy; from the Greek *autos* = self, and *nomos* = norm (Varela, 1979; Ruiz-Mirazo and Moreno, 2004; Di Paolo, 2004; Kauffman and Clayton, 2006). Although the origins of this approach can be traced back to the works of Aristotle and Kant (his *Critique of Judgement*), it was through the relatively modern development of theoretical biology and the physics and chemistry of far-from-equilibrium systems that it entered the scientific discourse. The contemporary conception of the organisational approach contends that norms are to be found as conditions of viability of the system, sometimes depicted in adaptive behaviour literature as a viability region (see Figure 1) or discussed as ‘viability constraints’ (Ashby, 1952; McFarland, 1999; Aubin et al., 2011). A closely related term is that of *precariousness* (Jonas, 1966, 1968; Weber and Varela, 2002; Di Paolo, 2005; Barandiaran et al., 2009), related, but not identical to the notion of “being far from thermodynamic equilibrium” when the system is a chemical or metabolic system (Ruiz-Mirazo and Moreno, 2004). The idea is that natural agents are organisms (i.e. living systems) that stand always in precarious conditions: if they don’t actively regulate their interaction with their environment (e.g. find food or a lower temperature) they will perish, since they exist in a continuous need of thermodynamic exchange with their environment. This precariousness is meant to form the basis of the normative character of behaviour: the system must actively seek to compensate its inherently decaying organization.

In a key paper where the theory of autonomy (in particu-

lar the autopoietic tradition) is complemented and expanded with Ashby's framework for adaptive behaviour, Di Paolo (2005) defined adaptivity (in relation to agency) as:

“a system's capacity, in some circumstances, to regulate its states and its relation to the environment with the result that, if the states are sufficiently close to the boundary of viability, 1. tendencies are distinguished and acted upon depending on whether the states will approach or recede from the boundary and, as a consequence, 2. tendencies of the first kind are moved closer to or transformed into tendencies of the second and so future states are prevented from reaching the boundary with an outward velocity.”

Di Paolo's definition of adaptive agency could be explicitly modelled and formalized. However, most of the models that have been developed with similar approaches have failed to address two blind spots: (1) viability boundaries appear as given or defined from without and the models focus on how to shape adaptive dynamics to maintain the trajectories of essential variables within those boundaries; (2) as a consequence, the relationship between the organismic dynamics that define the boundaries and the dynamics that control adaptive behaviour remain decoupled. In previous work (Egbert et al., 2009, 2010b) we have explored the relations between the viability boundary determining metabolic dynamics and the dynamics that drive organismic behaviour. A further problem remained however: although the boundaries of viability were directly linked to the modelled system, they were only defined by the system in a relatively trivial way. The boundaries of our models and similar efforts by others (see e.g. Ruiz-Mirazo and Mavelli, 2008) were the result of rough physical magnitudes: disappearance of the protocell due to complete lack of catalysts or bursting disintegration of the protocell marked by the upper limit of the tension of the membrane. The boundaries were not emergent from interactions between system processes in the holistic system-interdependent manner that characterizes integrity and systemic identity in real organism. In our previous models viability boundaries equated to absence of the system (i.e. total disintegration or zero quantity of its constituent elements). But, in natural systems, the limits of viability do not map with the physical disintegration of a system (Figure 2A), but rather with the loss of the capacity of the system to sustain itself. To lose viability is not to disappear altogether but to cross a much more subtle boundary after which the maintenance of life becomes impossible (Figure 2B). This boundary is the result of the dynamic organization of the system and, as we shall see, it defines a norm that behavioural patterns need to satisfy in order to be adaptive.

In this paper we model a minimal protocell-like system whose metabolic dynamics define an emergent viability boundary. For fixed concentrations of available resources, we can plot a bifurcation diagram of the chemodynamics

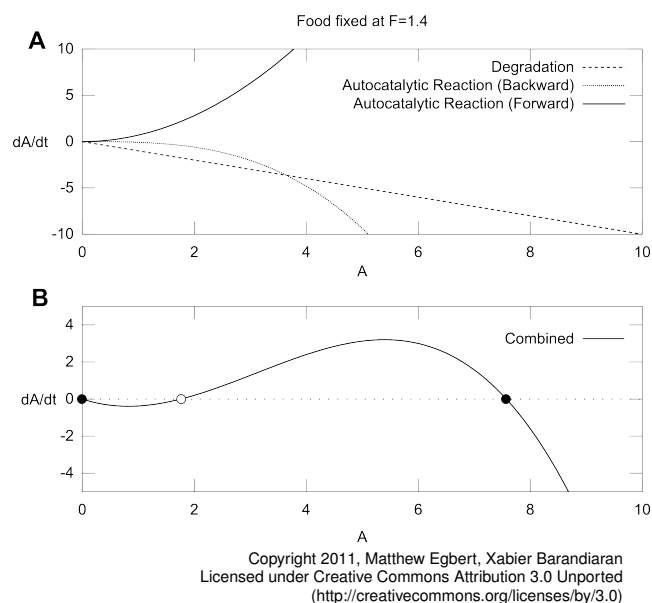


Figure 3: A) The influences of the forward and backwards flow of the autocatalytic reaction and degradation upon the concentration of A. B) The combined influence of the chemical reactions and degradation upon the concentration of A given a fixed concentration of  $[F] = 1.4$ .

that indicates the intrinsic boundaries of viability of the system. Different viability regions can be identified and the adaptive norms of the system clearly defined and quantified.

We then couple a gradient climbing behavioural mechanism to the metabolic dynamics. We show that in this metabolism-behaviour coupled system, the behaviour of the system can be directly mapped into the viability space of the simulated agent and it is possible to explicitly show and quantify how the system is adaptive for and by itself.

## Model

### Minimal metabolism

The metabolic organisation of self-production is one of the most fundamental properties of living systems and has been studied as such by many (Kauffman and Farmer, 1986; Kauffman, 2003; Varela, 1979; Ruiz-Mirazo and Moreno, 2004). In creating and maintaining themselves, living systems define their own viability constraints—the necessary and sufficient conditions for their continued existence. Thus, for the present work, metabolism is particularly relevant because it captures precisely what we wish to study. In its minimal and essential form it suffices to model metabolism as the self-production of a chemical network through the transformation (by the network) of available resources into constituents of the network. In previous work (Egbert et al., 2010a,b, 2009) we have modelled these kinds of systems in more detail, but here we abstract the system into two categories of components that we use to approximate a more

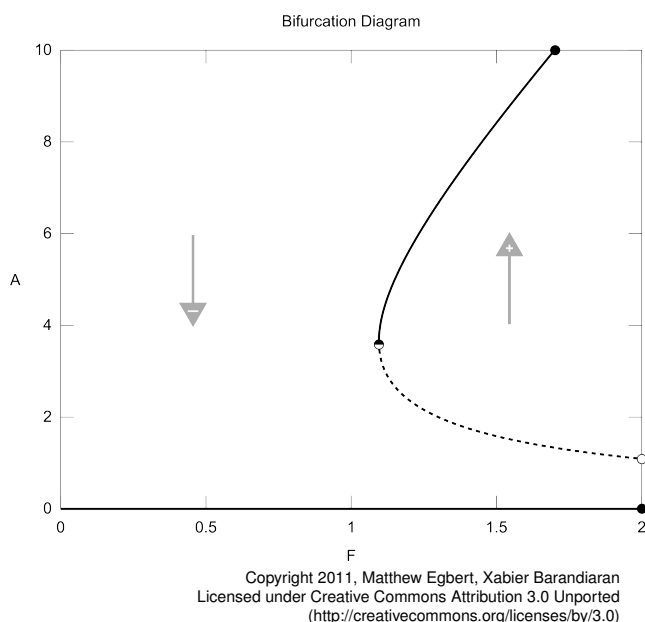


Figure 4: The system has a single stability at  $[A] = 0.0$  for low concentrations of  $F$ . At  $[F] \approx 1.1$  the system bifurcates, and for concentrations of  $F$  greater than this value, the system has two stable equilibria—the “living stable equilibrium” (where  $[A] > 0.0$ ) and “dead” (where  $[A] = 0.0$ ) and one unstable equilibrium, the viability boundary.

complicated metabolic system. These categories are ‘food’ or resource reactant(s) ‘ $F$ ’ and metabolites, i. e. members of the autocatalytic set, ‘ $A$ ’. Note that this approximation can be read as a higher order description of a more complicated system where  $A$  might capture an order parameter of a complex network of reactions among multiple molecules. In fact a recent and more complex model by Piedrafito et al. (2010) can be taken as dynamically similar to the present one, although it has a higher number of metabolites and catalysts and addresses other theoretically relevant properties (such as catalytic closure—which despite its relevance for the overall project of defining life and agency we have decided to leave aside for the specific purpose of the present paper).

We approximate the global dynamics of a more complex network according to the following reaction in which two members of the autocatalytic set interact with  $F$  to produce a third member of the autocatalytic set.



Note that the arrow is bidirectional, meaning that the reaction can occur in either direction, as is the case for all chemical reactions. A rate constant is associated with each direction (forward and backward) of the reaction  $k_b = 0.45$ ,  $k_f = 1.0$ . In addition to this autocatalytic reaction,  $A$  is subject to degradation into lower energy chemicals that are assumed to have no subsequent effect on the reaction and

are therefore not modelled. The combined influence of the forward and backward autocatalytic chemical reaction and the degradation are simulated by the following differential equation in which the degradation constant  $k_d = 1.0$ .

$$\dot{[A]} = \frac{-k_b[A]^3}{6} + \frac{k_f[F][A]^2}{2} - k_d[A]$$

## Metabolism-based chemotaxis

To study how behaviour can be sensitive to the viability boundary, we couple the metabolism to a simple stochastic gradient-climbing behavioural mechanism known as “run or tumble”. The run and tumble behaviour is inspired by the behaviour observed in *Escherichia coli* and other bacteria, that achieve chemotaxis through probabilistic modulation of two behaviours, “running” where the organism moves in a roughly straight line and “tumbling” where the organism chooses a new orientation at random. The mechanism modelled here is a form of *metabolism-based-chemotaxis*, meaning that no specific sensor nor chemical pathway is required to modulate behaviour; instead metabolism itself affects the behavioural probabilities so as to modulate the probability of tumbling (see Egbert et al., 2010b).

We have employed this coupling of metabolism and behaviour in previous papers to study the adaptability that such a coupling provides (Egbert et al., 2010b) and the possibility that an interaction between metabolism, behaviour and evolution can facilitate adaptive evolution of populations of protocells (Egbert et al., 2010a, 2011). Here we study how such a behavioural mechanism influences trajectories along the viability space.

In this case, the simulation of metabolism-based behaviour works as follows. The agent is considered to always be in a default state of running (moving in a straight line)  $\dot{x} = k\cos(\alpha)$ ,  $\dot{y} = k\sin(\alpha)$ . Tumbling occurs probabilistically with a likelihood that is modulated by the change in the concentration of  $A$ . If, since the previous iteration,  $[A]$  has decreased, the organism will tumble—i. e. a new orientation will be chosen from a flat distribution ( $\alpha = \text{rnd}[0..2\pi]$ ). Otherwise, the agent will continue running. A tumble inhibits any further tumbling for 5 iterations.

This particular form of metabolism-based run tumble mechanism is a highly simplified approximation of the “derivative” method used by *E. coli* and that simulated in (Egbert et al., 2010a, 2011) that compares the current metabolic rate to its rate a few moments previous. A decrease in metabolic rate indicates a worsening situation and increases the chance of a reorientation of the organism. In this way, the organism performs a simple but highly effective and surprisingly adaptive (Egbert et al., 2010b) behavioural strategy that can be captured by the anthropocentrism “If things are going well, I’ll keep going in this direction that I’ve been heading, otherwise, I’ll go somewhere else.”



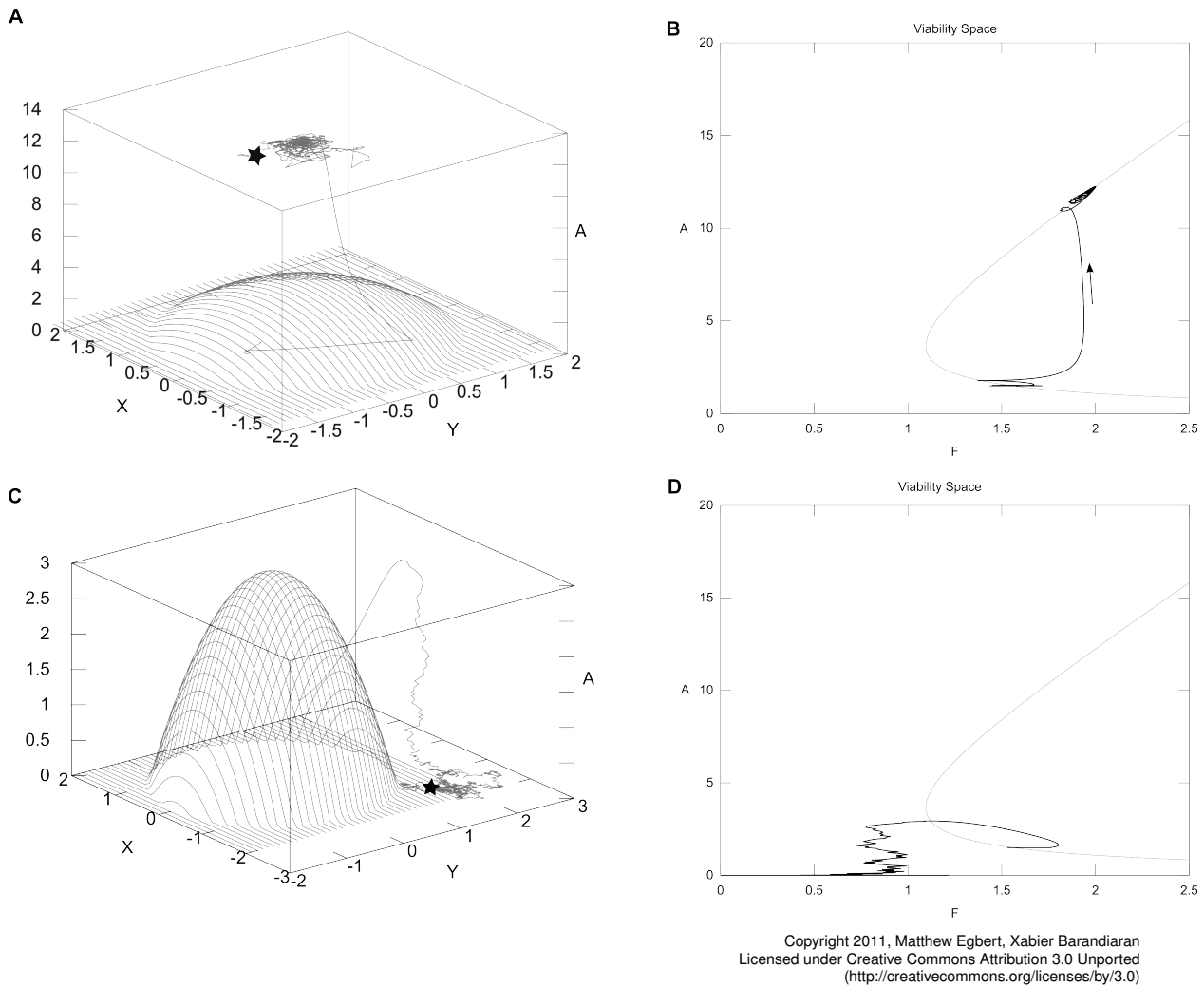


Figure 5: Paths taken by a successful chemotactic agent (top plots) and an unsuccessful agent (bottom plots). The left plots indicate the path of the agents in space plotted against  $[A]$ . The surface at the bottom of the image indicates the concentration of  $F$  in the environment. The right images show the path taken by the agents through viability space (see Figure 4). Initial oscillations around the viability boundary are eventually replaced by a trend up to the “living stable” equilibrium, thanks to the chemotactic motion.

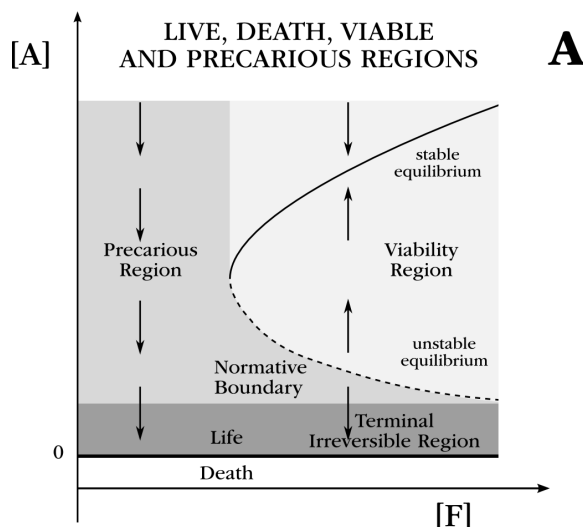
## Simulation results: metabolic and behavioural dynamics

### Metabolic dynamics: bifurcation line as viability boundary

We first consider the metabolic system independently of behaviour and study its dynamic for fixed concentrations of  $F$ . Intuitively, it is clear that with no food,  $[F] = 0$ , the system should be unable to maintain itself in the face of degradation. This is also the case for low concentrations of  $F$ . As we start to increase  $[F]$  however, the combined effect of its progressive disintegration and the forward and backward metabolic reactions of  $A$  leads to a bistable dynamic regime. The dynamic tendency of the three reactions and their combined effect for a fixed value of  $[F] = 1.4$  can be seen in

Figure 3A. It is clear from Figure 3B that this system has two stable equilibria, “death” at  $[A] = 0.0$  and “living stable” at  $[A] \approx 7.5$ , with an unstable equilibrium, the viability boundary at  $[A] \approx 1.8$ .

Analysis of the metabolic dynamics for different, fixed values of  $[F]$  gives us the bifurcation diagram in Figure 4. For  $[F] > 1.1$ , there is enough food to maintain a non-zero concentration of  $A$ . In this area of the parameter space, the system has two stable equilibria: “living stable” (where  $A > 0.0$ ) and “dead” (where  $[A] = 0.0$ ) and one unstable equilibrium, the *viability boundary* (the dashed line in Figure 4). Below the viability boundary, the system tends towards the “dead” equilibrium.



Copyright 2011, Matthew Egbert, Xavier Barandiaran  
 Licensed under Creative Commons Attribution 3.0 Unported  
 (<http://creativecommons.org/licenses/by/3.0>)

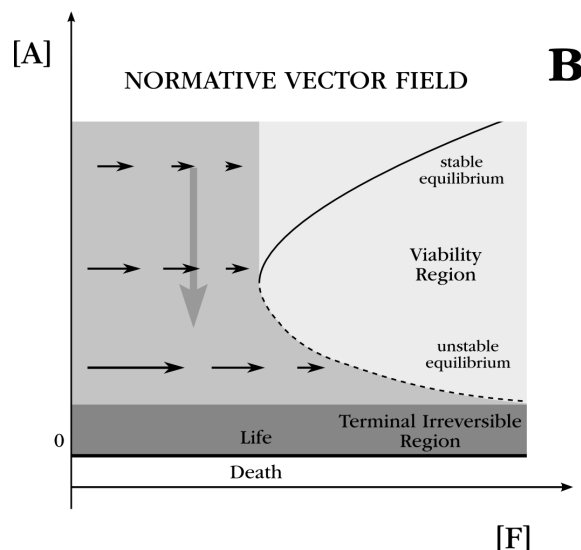
Figure 6: A summary of the regions of viability space: living, death, viable and precarious. See main text for further explanation.

### Chemotactic behaviour

Figure 5 shows the trajectories of two different agents using the metabolism-based mechanism. The left-hand figures shows motion in space plotted against  $[A]$ , the concentration of the autocatalyst. The right-hand figure shows the trajectory of the agent in “viability space” i.e. the same space as shown in the bifurcation diagram in Figure 4. The top images are for an agent that succeeded at performing chemotaxis. The lower images are the same, but for an agent that has had “bad luck” and the stochastic gradient climbing mechanism has failed.

### Model interpretation and discussion: Agency, precariousness, norms and adaptivity

This simple model suffices to satisfy a minimal requirement of normative behaviour, in that it generates a viability space where, living, viable, precarious, irreversible-terminal and death regions can be clearly identified. These are highlighted in Figure 6. The “dead region” can be clearly seen as the zero concentration of the required metabolites (a complete disintegration of the system). A *viable region* is identified where, given a fixed supply of resources, the system will maintain itself, growing or shrinking until it reaches the “living stable” equilibrium. The arrows indicate the tendency of metabolic dynamics for different regions of the viability space. The viable region can be precisely defined for a range of the parameter  $[F]$  and a range of initial conditions  $[A]$  as the subregion of the living space where for each point the evolution of the system will tend toward the stable living equilibrium. The unstable equilibrium at the bottom of the



Copyright 2011, Matthew Egbert, Xavier Barandiaran  
 Licensed under Creative Commons Attribution 3.0 Unported  
 (<http://creativecommons.org/licenses/by/3.0>)

Figure 7: The idea of the ‘normative field’ in the precarious region – the effects of behaviour as efforts to move the system into the region of viability.

viable region defines a lower boundary of viability below which, the system tends toward death. For small values of  $[A]$  and  $[F]$  we can distinguish a *precarious* region (medium grey area in Figure 6), where the system is still alive but will tend to die if the parametric condition  $[F]$  is kept constant, but could still recover if  $[F]$  is appropriately modulated. Underneath the precarious region a terminal-irreversible region can also be distinguished (dark grey area in Figure 6). If  $[A]$  falls in this region the system will be “alive” for some time, but will irreversibly die (given a certain limit of  $[F]$  increase, defined e. g. by diffusion).

We can now introduce the notion of a *normative vector field* defined by the minimal constant increase of  $[F]$  that is required at each point of the precarious region in order to move the state of the system into the viable region before the system reaches the terminal-irreversible region. Figure 7 is meant to illustrate this field: if the values of  $[F]$  and  $[A]$  are low (bottom-left side of the figure) the required increase of  $[F]$  is very big since the tendency of  $[A]$  will soon push the system to the terminal-irreversible region. If the concentration of  $F$  is low but there is a lot of  $A$  the required constant increase in  $[F]$  is low because the system has sufficient time to reach the viability boundary before the tendency to die becomes irreversible. Since  $[F]$  can be modulated by behaviour (provided that the environment displays a gradient of  $[F]$ ) a sense of *normative agency* can be precisely defined for every state of the system in the precarious region: the amount of increase of  $[F]$  that behaviour should achieve to compensate for its precariousness, that is the required movement in space that increases available  $[F]$  in accordance with

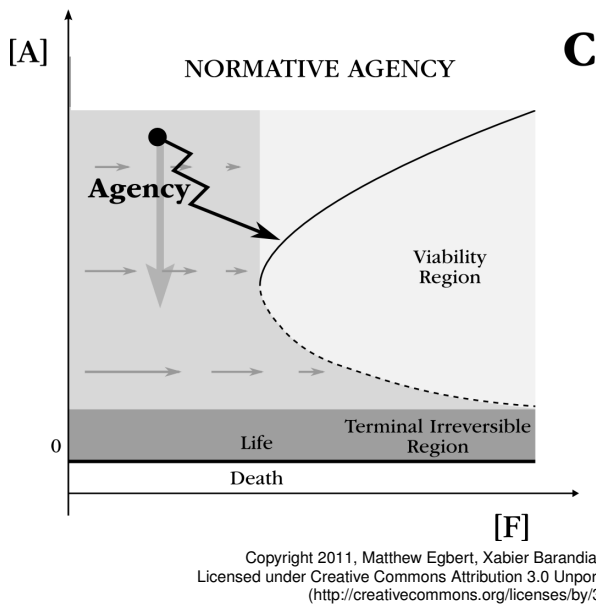


Figure 8: The effect of agency, idealised in this figure, but also seen in Figure 5. See main text for further explanation.

the normative field. Note that the system can fail to meet the norm, i.e. to adapt, for a variety of causes (e.g. because there is not enough  $[F]$  in the environment, because it cannot move sufficiently fast or does not manage to move up the gradient —like the case of the experiment shown in Figure 5-bottom). And yet the action can be said to be in accordance with the norm if it positively correlates with the normative field.

Agency can thus be clearly defined as the behavioural modulation that positively correlates with the normative field (which shall, given the appropriate environmental conditions, bring the system to its viable region). Figure 8 illustrates this point. To further illustrate this idea we examined an agent with a “perfect” gradient climbing mechanism that always moves directly up-gradient with a constant velocity. (Removing stochasticity from the behaviour in the model makes some of the dynamics easier to visualize.) Figure 9 plots one such “perfect gradient climber” with the same initial values of  $[A]$  but different distances from the peak of  $F$  gradient. We can see how the agent repeatedly moves from the precarious region back into the viability region, except for very low values of  $[F]$  for which the system, despite its behavioural modulation of  $[F]$  fails to reach the viability region and perishes – as the behavioural mechanism is insufficient to compensate.

## Conclusions

To conclude, we state that *for autonomous agency* (that is agency in relation to self-generated norms) *to take place the overall global constitutive dynamics of the system* (its self-maintaining organizational dynamics) *should at least* (that

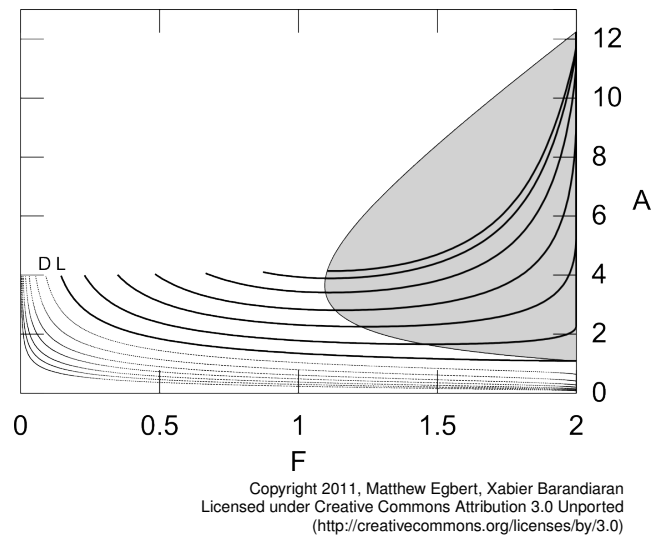


Figure 9: Trajectories of the system in the viability space as a function of its gradient climbing behaviour. Solid lines indicate trajectories that lead to the living stable equilibrium, dotted trajectories (despite behavioural influence) tend to death.

is minimally) *display an intrinsic topology with a viability boundary* (with the form of a bifurcation) *that defines a precarious region where behaviour can compensate for a death-ward tendency*. Arguably, it is only in relation to the intrinsically determined normative field that behaviour can be properly identified as adaptive and constitute a clear instance of natural normative agency.

The present model benefits from its low dimensionality in that it is easier to understand, but it also suffers, perhaps, from being over simplified in that there is really only two ways that the system can vary. Real organisms are of course much more complex and would display a multi-dimensional normative field and viability boundaries or surfaces. We are working on a more detailed model of a system similar to that described here in which the metabolism and behavioural mechanisms are more explicitly modelled (using more reactants and reactions). This will allow us to explore a greater variety of perturbations to system “health” as well as ways for the system to be sensitive (and therefore responsive) to its own viability. Another expansion of this work that we are considering is the quantification of a normative vector field and the formalization of the notion of positive correlation with it.

## Acknowledgements

Xabier E. Barandiaran holds a Postdoc with the FECYT foundation ([www.fecyt.es](http://www.fecyt.es)), and he is funded by Programa Nacional de Movilidad de Recursos Humanos del MEC-MICINN ([www.micinn.es](http://www.micinn.es)), Plan I-D+I 2008-2011, Spain. XEB also acknowledges funding from “Subvencion General

a Grupos de Investigacion del sistema universitario vasco. Grupo Filosofia de la Biologia” from Gobierno Vasco IT 505-10.

## References

- Ashby, W. R. (1952). *Design for a Brain: the origin of adaptive behaviour*. J. Wiley, London, 2nd edition.
- Aubin, J., Bayen, A., and Saint-Pierre, P. (2011). *Viability Theory: New Directions*. Springer, 2nd edition. edition.
- Barandiaran, X. and Moreno, A. (2008). Adaptivity: From metabolism to behavior. *Adaptive Behavior*, 16(5):325–344.
- Barandiaran, X. E., Di Paolo, E. A., and Rohde, M. (2009). Defining agency: Individuality, normativity, asymmetry, and spatio-temporality in action. *Adaptive Behavior*, 17(5):367–386.
- Burge, T. (2009). Primitive agency and natural norms. *Philosophy and Phenomenological Research*, 79(2):251–278.
- Christensen, W. D. and Bickhard, M. H. (2002). The process dynamics of normative function. *The Monist*, 85(1):329.
- Di Paolo, E. A. (2004). Unbinding biological autonomy: Francisco varela’s contributions to artificial life. *Artificial Life*, 10(3):231–233.
- Di Paolo, E. A. (2005). Autopoiesis, adaptivity, teleology, agency. *Phenomenology and the Cognitive Sciences*, 4(4):429–452.
- Egbert, M. D., Barandiaran, X. E., and Di Paolo, E. A. (2010a). Behavioral metabolism: Metabolism based behavior enables new forms of adaptation and evolution. In *Artificial Life XII Proceedings of the Twelfth International Conference on the Simulation and Synthesis of Living Systems*, pages 213–220. MIT Press.
- Egbert, M. D., Barandiaran, X. E., and Paolo, E. A. D. (2010b). A minimal model of Metabolism-Based chemotaxis. *PLoS Comput Biol*, 6(12):e1001004.
- Egbert, M. D., Barandiaran, X. E., and Paolo, E. A. D. (2011). Behavioral metabolism: The adaptive and evolutionary potential of metabolism-based chemotaxis. Submitted, March 2011.
- Egbert, M. D., Di Paolo, E. A., and Barandiaran, X. E. (2009). Chemo-ethology of an adaptive protocell: Sensorless sensitivity to implicit viability conditions. In *Advances in Artificial Life, Proceedings of the 10th European Conference on Artificial Life, ECAL*, pages 242–250. Springer.
- Frankfurt, H. G. (1978). The problem of action. *American Philosophical Quarterly*, 15(2):157–162. ArticleType: primary\_article / Full publication date: Apr., 1978 / Copyright 1978 North American Philosophical Publications.
- Jonas, H. (1966). *The Phenomenon of Life. Toward a Philosophy of Biology*. Chicago-London.
- Jonas, H. (1968). Biological foundations of individuality. *International Philosophical Quarterly*, 8(2):231–251.
- Kauffman, S. (2003). Molecular autonomous agents. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 361(1807):1089–1099.
- Kauffman, S. and Clayton, P. (2006). On emergence, agency, and organization. *Biology and Philosophy*, 21(4):501–521.
- Kauffman, S. and Farmer, J. (1986). Autocatalytic sets of proteins. *Origins of Life and Evolution of Biospheres*, 16(3):446–447.
- McFarland, D. D. (1999). *Animal Behaviour: Psychobiology, Ethology and Evolution*. Longman, 3 edition.
- Millikan, R. G. (1989). In defense of proper functions. *Philosophy of Science*, 56(2):288–302. ArticleType: research-article / Full publication date: Jun., 1989 / Copyright 1989 Philosophy of Science Association.
- Piedrafitra, G., Montero, F., Morn, F., Crdenas, M. L., and Cornish-Bowden, A. (2010). A simple Self-Maintaining metabolic system: Robustness, autocatalysis, bistability. *PLoS Comput Biol*, 6(8):e1000872.
- Ruiz-Mirazo, K. and Mavelli, F. (2008). On the way towards ‘basic autonomous agents’: Stochastic simulations of minimal lipid-peptide cells. *Biosystems*, 91(2):374–387.
- Ruiz-Mirazo, K. and Moreno, A. (2004). Basic autonomy as a fundamental step in the synthesis of life. *Artificial Life*, 10(3):235–259.
- Silberstein, M. and Chemero, A. (2011). Dynamics, agency and intentional action. *Humana Mente*, 15:1–19.
- Skewes, J. and Hooker, C. (2009). Bio-agency and the problem of action. *Biology and Philosophy*, 24(3):283–300.
- Varela, F. J. (1979). *Principles of biological autonomy*. North Holland New York.
- Weber, A. and Varela, F. J. (2002). Life after kant: Natural purposes and the autopoietic foundations of biological individuality. *Phenomenology and the Cognitive Sciences*, 1(2):97–125.

# On the Emergence of Structure in Behaviours Evolved Through Embodied Imitation in a Group of Robots

Mehmet D Erbas<sup>1</sup> and Alan FT Winfield<sup>1</sup>

<sup>1</sup>Bristol Robotics Laboratory, University of the West of England, Bristol, UK  
mehmet.eras@brl.ac.uk

## Abstract

This paper describes research in which we model social interactions between artificial agents using real robots. We show that variations that arise from embodiment allow certain behaviours, those that are more robust to the processes of embodied imitation, to emerge and evolve during multiple cycles of imitation. We test 3 memory strategies: no memory, limited memory and unlimited memory, and experimental results appear to show that with limited memory, those behaviours are more likely to become dominant within the robots' collective memory.

## Introduction

Social learning, which enables individuals to learn from each other, is a powerful mechanism in social animals, including humans. An important form of social learning is imitation, in which an individual observes and replicates another's actions. Imitation has been widely studied both by biologists and psychologists; biological research on imitation mostly focusses on its adaptive value for the organism, whereas psychologists are largely interested in the function of imitation and the mechanisms in which it plays a part (Zentall, 2001). There is continuing debate on the definition of imitation and whether it is unique to humans but what is not in doubt is that imitation clearly serves an important role in the development of social cognition in humans. For example, Dautenhahn et al reported that human babies are born with the ability to imitate a wide range of behaviours, including mouth opening and tongue protrusion (Dautenhahn et al., 2003). Meltzoff and Moore (Meltzoff and Moore, 1992) stated that human infants use imitation to enrich their understanding of people and their activities. Through imitation, humans are able to become part of a very complex social environment: human society. Imitation has also been seen as an important facet of cultural transmission; Dawkins argued (Dawkins, 1976) that imitation is a prerequisite for the evolution of culture, as it allows transmission of behaviours, with variation, between individuals.

The study of imitation in robotics has received cross-disciplinary attention in recent years. In the context of

robotics research, Bakker and Kuniyoshi (Bakker and Kuniyoshi, 1996) defined imitation thus: "Imitation takes place when an agent learns a behaviour from observing the execution of that behaviour by a teacher". This definition hints at how imitation is implemented and is used in most robotics research. Skill acquisition by human or robot demonstration has been widely investigated ((Scassellati, 1999); (Mataric, 2000)). This approach holds the promise that we may be able to overcome the necessity to program every behaviour a robot may need to perform, as the robot can learn new behaviours through observing demonstrations of those behaviours. However, as stated above, as well as supporting skill transmission between individuals, in human society, imitation has a social dimension, allowing individuals to become part of a social community. Alissandrakis et al. (Alissandrakis et al., 2004) stated that imitation may serve as a stepping stone towards the development of social cognition in artificial agents as it can form social integration with other artificial agents or with humans. Imitation research in robotics might also usefully address the question of how culture emerges and evolves as a novel property in groups of social animals. In (Winfield and Erbas, 2011) we introduce embodied imitation as a method for modelling the emergence of behavioural 'traditions' in social agents.

There has been some work examining the social dimension of imitation in robotics. Steels and Kaplan (Steels and Kaplan, 2001) stated that social learning can play a crucial role in initiating a humanoid robot into a linguistic culture. He used methods such as initiating open-ended dialogues among humans and robotic agents, in which social learning could be embedded. Billard (Billard, 1999) claimed that imitation can be used to enhance autonomous robots' learning of communication skills. The sharing of a similar perceptual context between the imitator and demonstrator can create the necessary social context in which language can develop. Billard devised some experiments in which robotic agents were able to learn a proto-language by using imitation to match their environmental perceptions with observed actions. In this paper, we aim to show that by sharing a similar perceptual context, agents involved in multiple cycles of imitation

can – in a sense – agree on the structure of the information that can best be transferred by imitation (that is, what can be imitated). Multiple robots are programmed to observe and imitate each other's movement patterns and the imitated behaviours undergo multiple cycles of copying, in which they mutate because of noise and uncertainties in the real robots' sensors and actuators. We observe that some movement patterns, which can be imitated with high fidelity, emerge and evolve in the group of real robots.

Alisandrakis et al. (Alisandrakis et al., 2004) developed the ALICE architecture (Action Learning via Imitation between Corresponding Embodiments) to address the problem of imitation between dissimilar embodiments. They examined the rules of synchronisation, looseness of perceptual matching and proprioceptive matching in a series of experiments in which robotic arms with variably-sized and numbered joints imitate each other. They showed that patterns can be transmitted between simulated robotic arms and variations occur during these replications because of heterogeneities between the arms. They argue that these variations provide the evolutionary substrate for culture, as new behavioural patterns may emerge and be transferred between agents. In this paper, we describe a series of experiments in which real robots observe and imitate each other's movement patterns. We show that even in an homogeneous group of real robots, variations occur during the imitation process that allow certain behavioural patterns to emerge and evolve during multiple cycles of imitation. These evolved behaviours can be copied with higher fidelity, as they are more robust to uncertainties in the real robots' sensors and actuators.

### Imitation in Robots

As stated above, we have used real robots to model the social interactions between agents. The motivation for using real robots rather than simulated agents or biological social entities for modelling is:

- Real robots, with their less than perfect perception and actuators, provide natural variations in the imitation process which allow new behaviours to emerge and evolve. Using simulated agents in a simulated environment, we would have to control the degree and types of heterogeneities and noise, but this may preclude any emergent processes that are a part of imitation; the level of emergence in a simulated environment would be limited to the level of variance that is artificially introduced.
- Data about the imitative activity, including the internal data and calculations of the robots, can easily be extracted and examined. This would not be the case if biological social entities (for example, people or monkeys) were used.
- The implementation of imitation on real hardware makes clear how theoretical assumptions and hypotheses regarding imitation can be operationalised.



Figure 1: A Linux-extended e-puck robot. The robots are fitted with coloured skirts, to enable them to 'see' each other. The yellow hat on top of the robot provides a matrix of pins holding unique patterns of reflective markers that allow the tracking system to identify and track each robot.

### Hardware Setup

The artificial agents used to model social interactions are *e-puck* miniature robots (Mondada et al., 2009), 7 cm in diameter and 5 cm in height. They are equipped with 2 stepper motors, two wheels of 41 mm diameter, 8 proximity sensors, a CMOS image sensor, an accelerometer, a microphone, a speaker and a ring of coloured LEDs. Their on-board battery provides 3 hours of autonomy. The robots are enhanced with a Linux extension board (Liu and Winfield, 2011) based on the 32-bit ARM9 micro-controller with the Debian/Linux system installed. The board has a USB extension port, used to connect a wireless network card, and is equipped with a MicroSD card slot. These additions to the standard e-puck robot offer increased processing power and increased memory. The robots are also fitted with coloured 'skirts' to enable them to see each other using their built-in image sensors. The experiments are performed in an arena measuring 3 m x 3 m. A vision-tracking system provides high-precision position tracking and a dedicated swarm server combines the data from the tracking system and the internal data from robots for later analysis. Each robot is also fitted with a tracking 'hat' which provides a matrix of pins holding unique patterns of reflective markers that allow the tracking system to uniquely identify and track each robot (Fig. 1).

### Movement Imitation Algorithm

In this research, a robot-to-robot movement imitation algorithm is implemented on the Linux extended e-puck robots.

Each robot is able to track and copy the other robot's movement patterns. Since we are interested in embodied imitation, the algorithm completely depends on the visual data coming from the image sensor of the robots; no other type of communication is allowed between the robots.

There are 3 main stages in the imitation algorithm:

- **Frame processing:** While observing captured visual frames, the observing robot tracks the movement of the demonstrator robot. As stated above, the robots are fitted with coloured skirts; by determining the size and location of the skirt on the demonstrator robot, the observing robot estimates the relative position of the demonstrator and stores this information in a linked list of positions. In this way, up to 5 frames per second are processed.
- **Data processing:** After the demonstrator's movement pattern is completed, the observer robot processes the linked list of positions using a regression line-fitting method to convert the estimated positions into straight line segments.
- **Pattern replication:** The straight line segments and their intersections are converted into a sequence of motor commands (moves and turns).

In this way, the observing robot replicates the pattern demonstrated by the demonstrator robot.

### Quality of Imitation

To quantitatively assess the fidelity of imitation (that is, the similarity between the original movement pattern and its copy), a quality of imitation function needs to be defined. Since each movement pattern consists of straight moves and turns, there are 3 components to each pattern that can be copied: the number of segments (straight moves), the length of each move and the angle (turn) between each consecutive move. Therefore, the overall quality of a copy can be calculated by separately estimating 3 quality indicators. The quality of move length,  $Q_l$ , between the original path  $O$  and its copy  $C$  is calculated as follows:

$$Q_l = 1 - \frac{\sum_m |l_m^O - l_m^C|}{\sum_m l_m^C} \quad (1)$$

where  $l_m$  is the length of move  $m$  that is to be compared. Here, the ratio is calculated of move length differences between the original pattern and its copy and the total move length of the copy. If the original movement pattern and its copy have different numbers of segments,  $N^O$  and  $N^C$  respectively, the sum is calculated only over the number of segments in the smaller:  $\min(N^O, N^C)$ . The quality of angle (turn) imitation similarly calculated as:

$$Q_a = 1 - \frac{\sum_m |a_m^O - a_m^C|}{\sum_m a_m^O} \quad (2)$$

where  $a_m$  is the turn angle following the move  $m$ . The quality of segment imitation simply compares the difference between the number of segments of the original pattern and its copy. It is calculated as:

$$Q_s = 1 - \frac{|N^C - N^O|}{N^O} \quad (3)$$

where  $N^O$  and  $N^C$  are the number of segments of the original path and its copy. The overall quality of imitation,  $Q_i$ , is a combination of 3 quality indicators:

$$Q_i = \frac{LQ_l + AQ_a + SQ_s}{L + A + S} \quad (4)$$

where  $L$ ,  $A$  and  $S$  are weighting coefficients.

To test the quality of imitation, a demonstrator robot is programmed to follow a sequence of straight line moves and turns that describes an equilateral triangle, while an imitator robot watches. Then, the imitator robot performs its copy of the demonstrator's pattern (Fig. 2). By comparing these two patterns, the quality of imitation is determined. The same scenario is repeated multiple times, with different distances between the robots. As shown in the figure, the best quality is achieved when the distance between robots is 1 m (Fig. 3). When the distance between robots is increased (to 1.5 m or more) the quality of imitation starts to degrade. This arises because the relative positional changes are estimated based on the size and location of the observed robot in the field of vision of the imitator robot. When the separation distance increases, the positional changes are harder to detect, as they cause smaller variations in the image of the observed robot. On the other hand, when the distance between robots is small (that is, 0.5 m or less), the demonstrator robot leaves the field of vision of the imitator robot many times, forcing the imitator robot to rotate itself each time and thus it may miss some turns of the demonstrator robot's trajectory while it is busy. Therefore, we have a separation range, between 0.5 m and 1 m, that is optimal for our vision based embodied imitation algorithm.

### Experiments

The notion of an imitation experiment is introduced to examine the effects of multiple cycles of imitation on the structure of the movement patterns that are being copied. During these experiments, 4 robots are placed in the arena, 1 m apart from each other (Fig. 4). Robots interact by copying each others' movement patterns using the imitation algorithm outlined in the previous section. Robots can be in one of two modes during the experiments: demonstrator or observer. When a robot enters demonstrator mode, it turns its LEDs on for 35 seconds to signal that it will start to demonstrate a movement pattern. During this period the demonstrator tries to grab the attention of one (or more) other robots. After that, the demonstrator robot turns its LEDs off and executes a movement pattern that consists



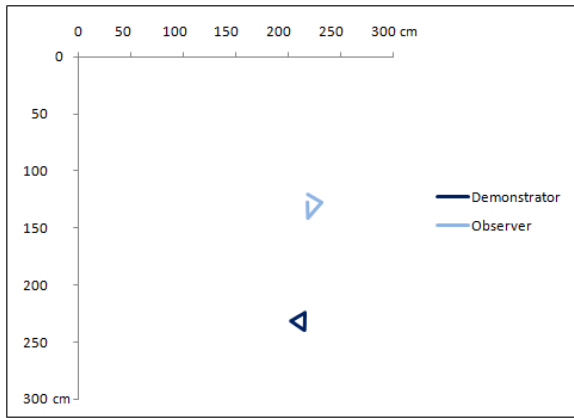


Figure 2: Plot of the trajectory of robots during an imitation run. The demonstrator robot moved in an equilateral triangular trajectory which was then copied by the observer robot. The robots were placed 1m apart.

of straight-line moves and turns. When execution is complete, the demonstrator robot blinks its LEDs for one second to signal ‘finish’. Then the demonstrator robot returns to its original start position and enters the observer mode. When a robot enters observer mode, it searches for a start signal by scanning the arena while rotating itself. When it detects a start signal, it focuses its attention on the demonstrator robot and waits for the demonstration to start. After completion of the demonstration, the observer robot records what it has observed and enters demonstrator mode. The finite state machine of the controller of the robots is shown in Fig. 5. At the start of the experiment, two of the robots are in demonstrator mode (Robots A and B) while the other two are in observer mode (Robots C and D). The experiment is left free-running as the robots change roles while imitating each other. All internal calculations and movement patterns of the robots are recorded for later analysis.

### Imitation with no memory

In the first set of experiments, the robots are able to remember only the most recent pattern they have observed; any newly-observed pattern replaces the previous one. Robot A is initialised with a square trajectory and Robot B is initialised with an equilateral triangle trajectory. Fig. 6 shows the pattern evolution map of an experiment in which 39 successful imitations were completed in approximately 20 minutes. In the figure, each node represents a pattern. If an arrow exists at a node, this means one of the robots executed that pattern and it was imitated by another robot. The new copy is at the end of the arrow. If the copy is high-quality, ( $Q_i \geq 0.85$ ) the node has a dark colour.

We first observe in this experiment that the original patterns deteriorate very quickly. At the beginning of the run, both robots (C and D), by chance, copied the square trajec-

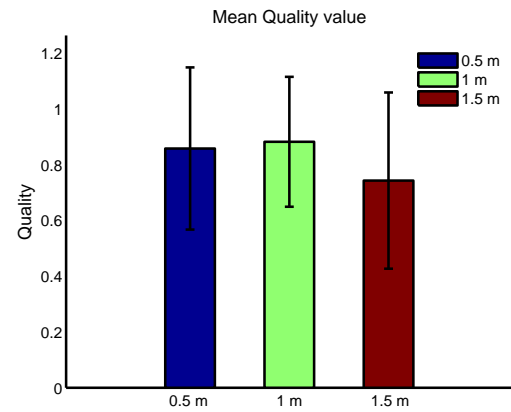


Figure 3: Mean quality of imitation ( $Q_i$ ) value with 95% confidence intervals calculated at different distances between robots. Each bar shows mean quality value over 20 cycles of imitation in which an equilateral triangle (each side 15 cm) movement pattern described by the demonstrator robot is copied by an imitator robot. For quality of imitation calculation, each quality indicator was given equal weight:  $L = A = S = 1$ .

tory and the triangular trajectory vanished from the experiment. The square trajectory also deteriorates rapidly, as any low quality copy can easily replace it. Because some bad copies missed turns, eventually the robots ended up with a pattern consisting of a single forward move. These low quality copies do not occur often but just one is sufficient to disrupt the evolution of the movement patterns. In this experimental run, all patterns after pattern number 22 consist of one single move without any turns. These single move patterns can be copied with high quality but we still observe some poor copies. We conclude therefore that when robots have no memory, evolution of the movement patterns is acutely sensitive to imitation errors.

### Imitation with unlimited memory

In the second set of experiments robots have unlimited memory so they save all patterns that they have observed. When they enter demonstrator mode robots randomly select, with equal probability, one of the patterns in their memory and demonstrate it. Once again Robots A and D are initialised with a square trajectory and Robots B and C are initialised with an equilateral triangle trajectory. Fig. 7 shows the pattern evolution map for an experiment with this setting. In this run, 55 successful imitations were completed in 30 minutes. We first observe in these experiments that – as we would expect – the original movement patterns are more likely to be preserved (with variation), as each newly-observed pattern is stored in memory. Low quality copies occasionally occur, but as they do not replace previously ob-

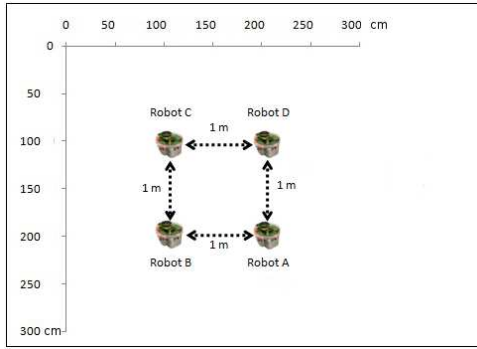


Figure 4: Each experiment presented in this section is performed in a 3 m by 3 m arena with 4 robots, placed 1 m apart and arranged as shown here. In all experiments, Robot A and Robot B are started in demonstrator mode while Robot C and Robot D are started in observer mode.

served patterns, these paths cannot easily become dominant. Second, we see that as patterns evolve during multiple cycles of imitation, some paths that are able to be copied with high quality emerge and propagate among robots. In this run, pattern 27 has this property. Fig. 8 shows the evolution of pattern 27. It is a descendant of the original equilateral triangle trajectory, and there are 5 intermediate copies between the original triangle and pattern 27. At each copy, the pattern is modified by the imitating robot. Finally pattern 27 emerges and a sharp increase in quality of imitation can be observed after this point ( $Q_i > 0.94$  for all of its descendants). What makes this pattern and its descendants easily copiable? First, short moves are more prone to error, as a small mistake in perception can cause them to vanish; a pattern that can be copied with high quality typically does not include short moves. Second, the length of each move varies at each subsequent copy. Although estimating the relative size and position of the demonstrator robot is straightforward image processing, it is error-prone, because of the relatively low resolution of each robot’s image sensor. A move directed towards or away from the observing robot can only be detected if it causes a perceptible change in the size of the demonstrator robot, i.e. a detectable change in number of pixels in the image of the demonstrator. At each copy, the observing robot stores what it infers from the demonstration, as perceived from its relative position and perspective. Therefore, the patterns tend to evolve into ones that can be more easily imitated. Fig. 9 shows pattern 27 and its descendants. As can be seen, there is a high level of similarity between these patterns. At the end of the run, pattern 27 and its descendants form a cluster of similarly-shaped patterns in the robots’ memories. Fig. 10 shows the average  $Q_i$  value for this experiment in comparison with the average  $Q_i$  value for the cluster formed by pattern 27’s descendants. As can be seen, although the distance between the robots is 1 m, the

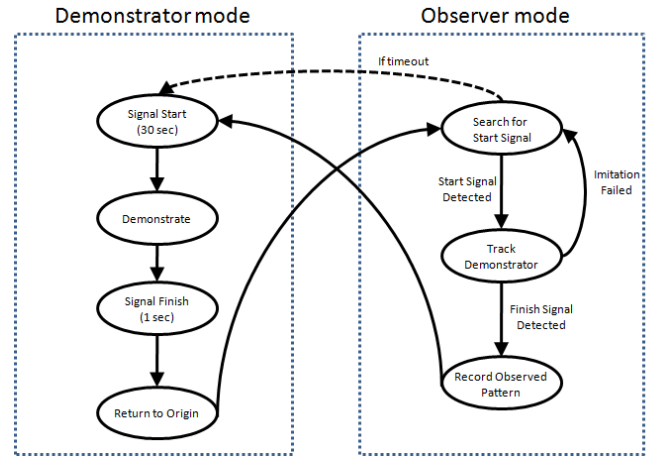


Figure 5: Finite state machine of the controller of the robots. The robots are programmed to copy each others’ movement trajectory as they keep changing their roles to demonstrator and observer. To prevent a deadlock with all the robots searching for a start signal, two of the robots (Robot A and Robot B) are programmed to time out and enter ‘Signal Start’ state after completing two complete scans of the arena in ‘Search for Start Signal’ state (the dashed arrow).

average  $Q_i$  value is slightly low; around 0.83. This can be explained by the fact that some low quality imitations occur during the evolution of patterns. A sharp increase in  $Q_i$  value can be observed after a pattern emerges that is more robust to uncertainties in the robot’s sensors and the imitation process: the average  $Q_i$  value for the cluster that is formed by the descendants of pattern 27 is 0.96.

### Imitation with Limited Memory

In the previous set of experiments, we showed that certain patterns, those that are more robust to uncertainties in the real robots’ sensors and actuators and the estimation process of imitation, can emerge and evolve during multiple cycles of imitation. As these emergent patterns can be copied with high quality, their descendants have similar, inherited characteristics. As a result, clusters of highly copiable patterns are formed in the memories of the robots. These clusters may grow larger with subsequent cycles of imitation if, by chance, members of these clusters are selected for demonstration. We now show that with a limited memory, these emergent patterns and their copies can become dominant. In the third set of experiments, an example run with limited memory is presented, in which an emergent pattern and its highly similar descendants become dominant. Here robots have a limited memory, in which they can store only the most recent 5 patterns observed. When the memory is full and a new pattern observed, the oldest pattern in memory is replaced with the new pattern. Fig. 11 shows the pat-

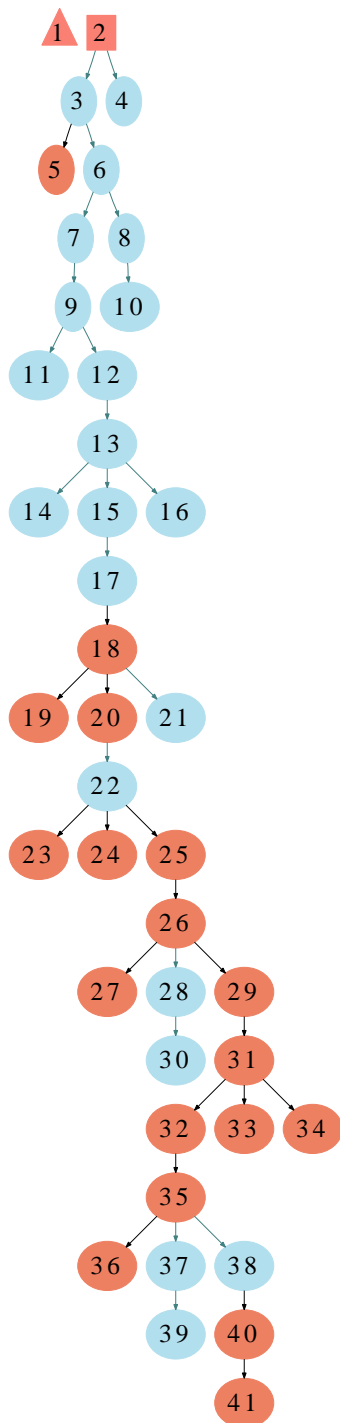


Figure 6: Pattern evolution map for a 4 robot experiment with no memory. Each node in the figure represents the demonstration of a movement pattern. If a pattern is demonstrated and imitated, the new copy of that pattern is linked to it by an arrow. For instance, pattern 2, the original square, was demonstrated by Robot A and was copied by two robots. The new (child) copies of pattern 2 are patterns 3 and 4. If the copy is of high quality (i.e.  $Q_i \geq 0.85$ ), then the node has a dark colour.

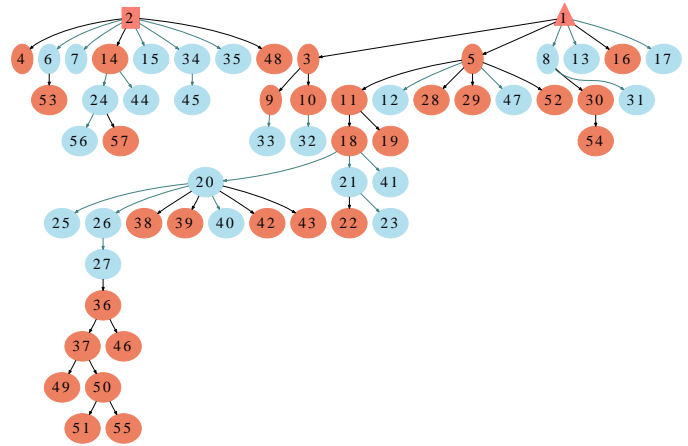


Figure 7: Pattern evolution map for a 4 robot experiment with unlimited memory. Initial movement patterns are a triangle (1) and a square (2).

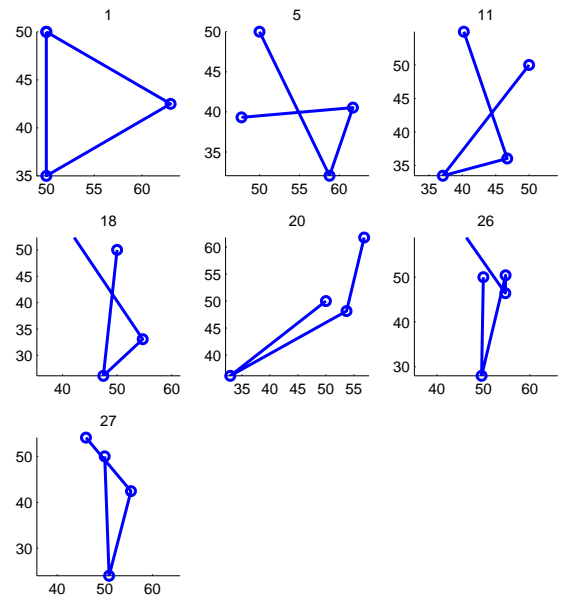


Figure 8: Evolution of pattern 27 in Fig. 7. Pattern 27 is a descendant of the original equilateral triangle pattern. By following the imitation links on the pattern evolution map for this experiment, we can see that there are 5 intermediate copies between the original triangle and pattern 27: the patterns numbered 5, 11, 18, 20, 26. All of these patterns, starting with the original triangle and ending with pattern 27, are shown here in order. All axis are marked in cm.

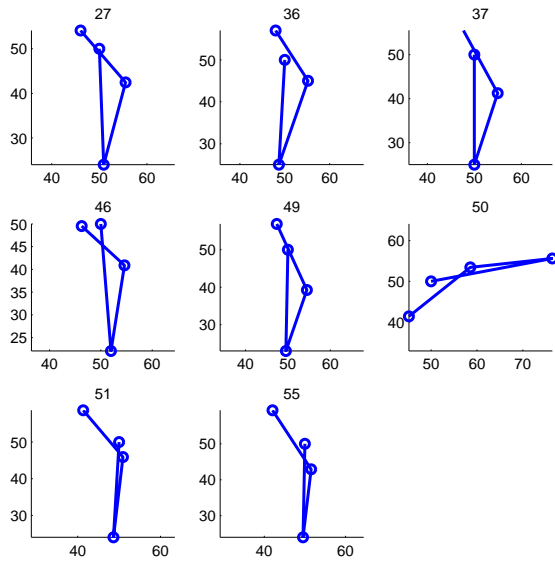


Figure 9: The descendants of pattern 27 in Fig. 7. Starting with pattern 27, its descendants (patterns 27, 36, 37, 46, 49, 50, 51, 55) are shown in order. All axes are marked in cm.

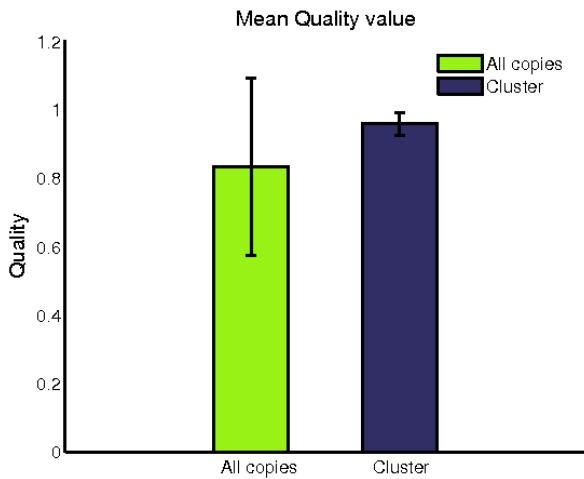


Figure 10: Average  $Q_i$  value for all imitation events in the experiment shown in Fig. 7 (All copies) and average  $Q_i$  value for the cluster formed by pattern 27’s descendants (Cluster), with 95% confidence intervals

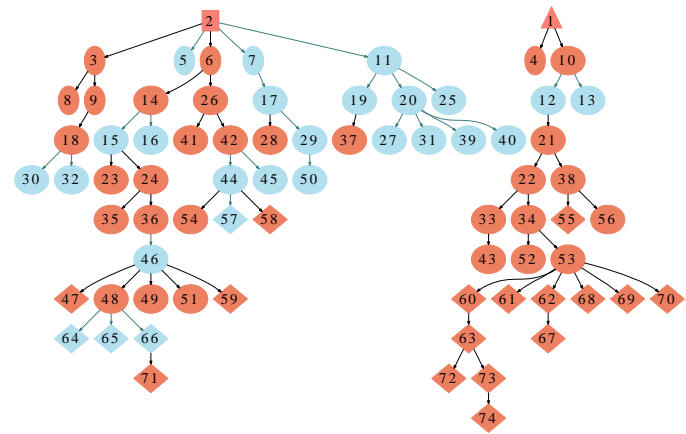


Figure 11: Pattern evolution map for experiment with limited memory. The 20 patterns in the memory of all 4 robots at the end of the experiment are highlighted as diamonds.

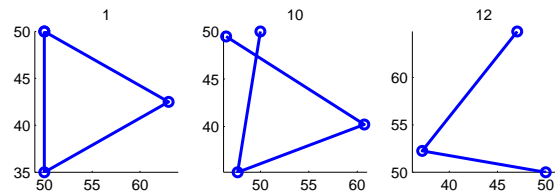


Figure 12: Evolution of pattern 12 in Fig. 11. There is an intermediate copy (10) between the original triangle and pattern 12. All axes are marked in cm.

tern evolution map from an experiment with these settings in which 72 successful imitations were completed in 60 minutes. In this run, a V-shaped pattern, pattern 12, emerged and all of its descendants are high quality copies. Fig. 12 shows the evolution of this path and Fig. 13 shows some of its high-quality descendants. At the end of this run, 12 of the 20 patterns in the memory of all 4 robots are descendants of this pattern. Since the robots randomly choose which pattern to demonstrate, there is now a 60% chance that one of the descendants of pattern 12 will be demonstrated again. Once it is selected and copied, the new copy is itself likely to be a high quality copy and so similar to pattern 12. This process will then increase the percentage of patterns in the memory that are similar to pattern 12. We conclude therefore that with limited memory, patterns robust to uncertainties that emerge are more likely to become dominant.

### Conclusion and Discussion

In this work, we have used real robots to model social interactions between artificial agents, in particular learning by imitation. We have shown that variations in the real robots’ sensors allow certain behaviours to emerge and evolve during multiple cycles of imitation. These evolved movement

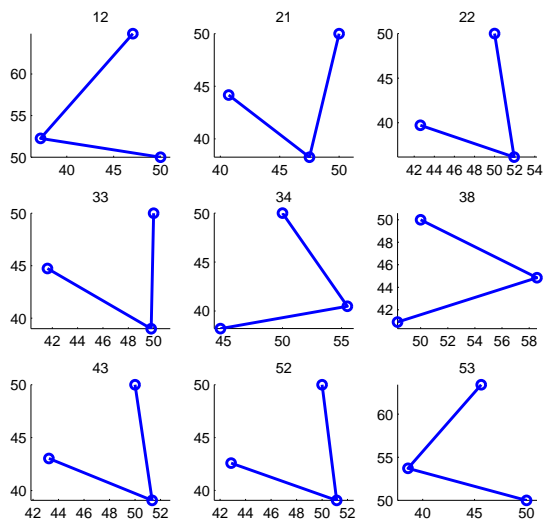


Figure 13: The descendants of pattern 12 in Fig. 11. Starting with pattern 12, some of its high-quality copy descendants (patterns 12, 21, 22, 33, 34, 38, 43, 52, 53) are shown in order. All axes are marked in cm.

patterns are more robust to the uncertainties of the real robot embodied imitation process and so they can be imitated with high fidelity. As the robots share a similar perceptual context and embodiment, they are able to – in effect – agree on the structure of the movement patterns that can be transferred between them.

We have experimentally tested three cases with different sizes of robot memories: no memory, unlimited memory and limited memory, in order to test the hypothesis that memory size will effect the likelihood of dominant movement patterns emerging. In the no memory case, the evolution of movement patterns is extremely sensitive to any instance of poor quality imitation, which means that the original movement patterns very quickly deteriorate. In the unlimited memory case, patterns emerge that can be easily copied but are less likely to then become dominant, as the number of patterns in the robots’ collective memory grows larger with each new imitation cycle. However, in the case with limited memory, these evolved patterns can become dominant if they and their descendants are, by chance, chosen for demonstration. For simplicity of analysis, we have carried out our limited memory experiments with a small memory size (5 patterns per robot). We conjecture that with a larger (but still limited) memory, multiple patterns that can be imitated with high fidelity can emerge and form clusters of similarly-shaped patterns in the robots’ collective memory. In this way, the robots can collectively evolve an ensemble of patterns that can be copied between them with high fidelity. Here the imitated patterns are not linked to a task or

an environmental context. However it seems possible, and testable using the embodied approach outlined in the paper, that associating imitation with behaviours that have utility could lead to the emergence of non-verbal communication between robots.

## Acknowledgements

This work is supported by EPSRC research grant: EP/E062083/1.

## References

- Alissandrakis, A., Nehaniv, C. L., and Dautenhahn, K. (2004). Towards robot cultures? learning to imitate in a robotic arm test-bed with dissimilar embodied agents. *Interaction Studies: Social Behaviour and Communication in Biological and Artificial Systems*, 5(1):3–44.
- Bakker, P. and Kuniyoshi, Y. (1996). Robot see, robot do : An overview of robot imitation. In *AISB96 Workshop on Learning in Robots and Animals*, pages 3–11.
- Billard, A. (1999). Imitation: A means to enhance learning of a synthetic proto-language in an autonomous robot. In *Imitation in Animals and Artifacts*, pages 281–311. MIT Press.
- Dautenhahn, K., Nehaniv, C. L., and Alissandrakis, A. (2003). Learning by experience from others - social learning and imitation in animals and robots. In Kühn, R., Menzel, R., Menzel, W., Ratsch, U., Richter, M. M., and Stamatescu, I. O., editors, *Adaptivity and Learning: An Interdisciplinary Debate*, pages 217–241. Springer Verlag.
- Dawkins, R. (1976). *The Selfish Gene*. Oxford University Press.
- Liu, W. and Winfield, A. F. T. (2011). Open-hardware e-puck linux extension board for experimental swarm robotics research. *Microprocessors and Microsystems*, 35(1).
- Mataric, M. J. (2000). Getting humanoids to move and imitate. *IEEE Intelligent Systems and their Applications*, 15(4):18–24.
- Meltzoff, A. N. and Moore, M. K. (1992). Early imitation within a functional framework: The importance of person identity, movement, and development. *Infant Behavior and Development*, 15(4):479–505.
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotocz, A., Magnenat, S., Zufferey, J. C., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *9th Conference on Autonomous Robot Systems and Competitions*, page 59:65.
- Scassellati, B. (1999). Knowing what to imitate and knowing when you succeed. In *AISB’99 Symposium on Imitation in Animals and Artefacts*, pages 105–113.
- Steels, L. and Kaplan, F. (2001). Aibo’s first words: The social learning of language and meaning. *Evolution of Communication*, 4(1):3–32.
- Winfield, A. F. T. and Erbas, M. D. (2011). On embodied memetic evolution and the emergence of behavioural traditions in robots. *Accepted for Journal of Memetic Computing*.
- Zentall, T. R. (2001). Imitation in animals: Evidence, function, and mechanisms. *Cybernetics and Systems*, 32:63–96.

# The role of energy in a stochastic model of the emergence of autocatalytic sets

Alessandro Filisetti<sup>1</sup>, Alex Graudenzi<sup>1</sup>, Roberto Serra<sup>1,2</sup>, Marco Villani<sup>1,2</sup>, Davide De Lucrezia<sup>1</sup>, and Irene Poli<sup>1,3</sup>

<sup>1</sup>European Centre for Living Technology, S.Marco 2940, 30124, Venice (IT)

<sup>2</sup>Department Communication and Economics, University of Modena and Reggio Emilia, Via Allegrì 9, 42100 Reggio Emilia (IT)

<sup>3</sup>Department of statistics, University Ca' Foscari of Venice, 30121 Venice (IT)  
alessandro.filisetti@ecltech.org

## Abstract

In most theories concerning the origin of life autocatalytic sets are supposed to play an important role in the phase transition between non-living and living matter. Although several theoretical models describe this phase transition, it is very hard to recreate the experimental conditions in wet lab. We here introduce a stochastic model of catalytic reaction networks with energy constraints, devoted to the study of the emergence of autocatalytic sets, in which some of the assumptions of the already existing model are relaxed in order to explore the possible reasons which make the emergence of autocatalytic cycles difficult or which make them unstable. Moreover, since living systems operate with a continuous exchange of matter and energy with the environment, we investigate the effects on the model behavior of changes in the rate of the energy intake.

## Introduction

The life as we know today is the result of billions of years of evolution and, even though the first forms of life were simpler than today, a certain degree of complexity was surely necessary in order to lead off the phase transition between non-living and living matter.

Although different scenarios for the onset of life have been proposed<sup>1</sup>, autocatalytic sets of molecules (ACSs) are considered of paramount importance to both extant biological systems and during the transition from non-living to living systems.

In the first case, ACSs represent the basic architecture of some of the most fundamental metabolic processes such as the citric acid cycle urea cycle, calvin cycle and beta-oxidation (Alberts et al., 2005), on the other hand, the

emergence of ACSs might have played a pivotal role in acquiring autonomy and homeostasis during the emergence of the first living systems (Ruiz-Mirazo and Mavelli, 2008) and they have been regarded ever since as the blueprint of primeval living systems (Fishkis, 2007; Ma et al., 2010).

Though the RNA world scenario, with particular regard to the role of the ribozymes (Gilbert, 1986; Talini et al., 2009), provides a plausible solution with respect to the prebiotic storage and replication of the information, it has been proven that template-dependent polymerization can occur only for relatively short nucleotides strands catalyzed by remarkably long RNAs (Bartel and Unrau, 1999; Bartel, 1999).

On the other hand, looking at the metabolism-first approach, the self-replication of only a single catalyst is plausible only within a very complex chemical system.

In his theory concerning the emergence of ordered structures and patterns of activation from disordered interactions (the so called “order for free” hypothesis (Kauffman, 1993)) Stuart Kauffman pointed out the idea that all that is needed is the presence of a set of molecules composed of a sufficient number of different molecular types in which each molecule catalyzes a step in the formation of one or more other molecules in the set; then the catalytic closure is reached if each molecule in the set is catalyzed by at least one other molecule of the set. Based on a combinatorial approach, Kauffman stated that the emergence of autocatalytic sets is inevitable when the molecular diversity reaches a certain threshold (Kauffman, 1986).

While the Kauffman initial approach is based on an analysis of the reaction graph, without taking into account the dynamics, there are several models in the literature that study autocatalytic systems from a dynamical point of view, such as those by Dyson (Dyson, 1985), Eigen and Schuster (Eigen and Schuster, 1977), Kauffman (Kauffman, 1986), Farmer and Bagley (Bagley and Farmer, 1992; Bagley et al., 1989), Jain and Khriشنا (Jain and Krishna, 1998), Lancet (Segre et al., 1998) and Kaneko (Kaneko,

<sup>1</sup>The main theoretical frameworks can be divided in the “gene first” approach, based on the template matching (Gilbert, 1986; Müller, 2006; De Lucrezia et al., 2007; Anastasi et al., 2007; Talini et al., 2009; Rios and Tor, 2009; Budin and Szostak, 2010), the “metabolism first” approach, based on the self-organization of the chemicals involved in (Oparin, 1924; Miller, 1953; Eigen and Schuster, 1977; Kauffman, 1986; Mossel and Steel, 2005; Lee et al., 1996; Saghatelian et al., 2001; Lifson, 1997) and the lipid-world (Segre et al., 1998)



2006).

Even if all these models predict the emergence of an autocatalytic set, observing it in a wet lab experiment remains a very difficult task. On the one hand, it is possible that the simplifications introduced by the in-silico models are unrealistic with respect to the extant biological systems but, on the other hand, the indications provided by the theory may be not correctly implemented in actual experiments.

In previous works (Fuchslin et al., 2010; Filisetti et al., 2011a,b, 2010) we introduced a novel stochastic model devoted to the study of the generic proprieties of catalytic reaction networks based on a particle description of the system, while in the present work we investigate the effects of the introduction of energetic constraints in the system.

Living systems cannot operate isolated from the environment and they need a continuous flow of energy and matter in order to be maintained far from the equilibrium. While the incoming flux of matter is necessary in order to feed the system with the elementary nutrients to be transformed in more complex molecules, energy is channeled into the construction of molecules whose constitutive reactions are energetically unfavorable. Energy is stored as chemical bonds in molecules called “carrier molecules”, which diffuse rapidly in the cell and thereby carry energy from places of energy generation to the reactions requiring energy to occur (Alberts et al., 2005).

While results concerning the influence of different composition of the incoming flux of matter have been presented in (Filisetti et al., 2010; Fuchslin et al., 2010; Filisetti et al., 2011a,b), here we focus on the role of the energy, some first indications can be found in (Fuchslin et al., 2010), within a system composed of both energetically unfavorable and favorable reactions.

It is important to remark that, coherently with the scientific approach typical of complex systems biology (Kaneko, 2006), we are not interested in investigating the specific nature of the chemicals present in our model, nor the particular interactions among them, but rather in the characterization of the dynamical behaviors emerging from the interaction of a set of chemicals and in the detection of possible generic properties of this kind of systems.

In section II the principal features of the models are presented while in section III we describe how the energy has been introduced in the stochastic model. In section IV we discuss some preliminary results of a set of simulations in which we study the influence of the amount of energy introduced in the system and, in the final section, conclusions and indications for further works are provided.

## Description of the model

An exhaustive description of the model can be found in (Filisetti et al., 2011a) and (Filisetti et al., 2010); we here summarize the principal features for a better comprehension of the paper.

Taking inspiration from the original work by Kauffman (Kauffman, 1986) the principal entities of the model are linear chains, species from now on, oriented from left to right, composed of the concatenation of letters from a finite alphabet, e.g.  $[A,B]$  or  $[A,G,C,T]$ .

Let  $X$  stands for the entire set of species and  $x_i, i = 1, \dots, N$ , representing each single species. In accordance with the stochastic nature of the model the total amount (quantity of *molecules*) of each species  $x_i$  is denoted by  $\hat{x}_i$ . Since the reactions take place in a well-stirred tank reactor with fixed volume, the relation between concentrations and quantities is straightforward.

The dynamics of the system is ruled by two different reactions, namely *condensation* and *cleavage*. By means of the former two species are concatenated in order to create a longer species (e.g.  $AB + BA \rightarrow ABBA$ ), whereas by means of the latter one specie is cut in order to create two shorter species (e.g.  $ABBA \rightarrow AB + BA$  or  $ABB + A$  or  $A + BBA$ ), in general given a species of length  $l$  there are  $2(l - 1)$  different cleavage products.

We assume that for spontaneous reactions the rate of the backward reactions is negligible with respect to that of the forward reactions (i.e. strongly negative  $\Delta G^\circ$ ).

Furthermore, since we are interested in the behaviors of catalytic reaction networks we assume that no reaction proceeds without the aid of catalysts, namely all reactions are characterized by a high energy barrier (i.e. activation energy) that would make them too slow to be observed in the absence of the correspondent catalysts.

The main novelty presented in this work is the explicit introduction of energy constraints, according to which some types of reaction require energy to proceed, as it will be described in the following section.

It is important to notice that the present version of the model neglects any catalysis provided by elements other than species belonging to the system, even though environmental catalysts, such as mineral clay, are thought to have played a relevant role in prebiotic synthesis (Ferris et al., 1996).

Given the number and the length of the species present in the system one can compute the overall number of conceivable reactions, including both cleavage and condensation, as

$$\hat{R} = \sum_{i=1}^N (L(x_i) - 1) + N^2. \quad (1)$$

where  $L(x_i)$  is the length of the  $i$ -th species and  $N$  is the



total number of species present in the system. An important assumption is that we consider an independent probability  $p$  that any species catalyses a random reaction, hence not all the  $\hat{R}$  conceivable reactions will occur, but only those that are actually catalyzed by some of the existing species.  $p$  turns out to be one of the key parameters of the model, since it rules the overall activity of the system by tuning the number of possible catalysts present in the reactor.

The dynamics is based on the well-known Gillespie stochastic algorithm (Gillespie, 1977, 2007) but, in order to speed up the computational performance, some of the processes are described by means of an approximated algorithm; in particular the ingoing and outgoing fluxes and, as we will see in the next section, the dynamics related to the introduction of the energy.

In accordance with the nature of the reactions, i.e. condensation and cleavage, we can summarize the reaction scheme as following:

- Cleavage:  $AB + C \rightarrow A + B + C$
- Condensation: (whole reaction:  $A + B + C \rightarrow AB + C$ )
  - Complex formation:  $A + C \rightarrow A : C$
  - Complex dissociation:  $A : C \rightarrow A + C$
  - Final Condensation:  $A : C + B \rightarrow AB + C$

where  $A$  and  $B$  are two generic substrates involved in a specific reaction,  $C$  is the specific catalyst for that reaction and  $A : C$  represents a temporary complex, which is necessary for the condensation process to happen.

One of the main features of the model concerns the possibility to create new species by means of the internal dynamics. The creation of new species leads to the creation of new reactions; on the other side, some species could also vanish. To maintain the consistency of the system in the case of reappearance of some of the vanished species, all the reactions are kept in memory.

Another important remark regards the emergence of competition and inhibition phenomena by means of the particle-based algorithmic approach, since the molecules involved in a specific reaction cannot be used in another one at the same time.

Notice that with regard to an asynchronous stochastic model such this, the question on the correct reaction graph to use is of fundamental importance. To this end we introduced three distinct reaction graphs, to be used according to the circumstances. In detail:

- The *possible reactions graph*: in which all the possible reactions at a certain moment of the simulation, including those that will not actually occur, are drawn.

- The *complete reaction graph*: in which all the reactions that occur at least once over the simulation time frame are conserved.
- The *actual reaction graph*: after defining a specific temporal window,  $W$ , only the reactions that occur within  $W$  at a specific time are kept in the graph, while the older ones are removed. Notice that the temporal window turns out to be a key parameter in the analysis of the system, since the detection of ACSs is made using this specific graph representation. In this way it is possible to define cycles even in a stochastic system with asynchronous update and, at the same time, to neglect the influence of very rare reactions on the overall dynamics.

## Introduction of the energy in the model

The first rationale at the base of the introduction of the energy within the model is that energy *does* exist in nature and, to a wide extent, it deeply affects the nature and the dynamics of biophysical and biochemical systems. With regard to our model of catalytic reaction network, both information and matter were present in the original description (Filisetti et al., 2010, 2011a), while energy was missing. Therefore, one of the major objectives of this work is to decipher whether and how energy actually influences the overall dynamical behavior. Moreover, we may hypothesize that the association of energy to some specific type of reaction could lead the system to novel complex behaviors, mostly in regard to the possible emergence of ACSs.

The general idea is to divide the possible reactions in two classes in accordance with the specific energetic constraints, namely *exergonic* and *endergonic* reactions. While exergonic reactions occur releasing energy, endergonic reactions require the presence of energy carrier molecules that release an amount of energy to some of elements involved in the reaction, otherwise the reaction cannot occur<sup>2</sup>.

For simplicity we assume that the exergonic reactions release energy in form of heat (in the present state of the model there is no coupling between exergonic and endergonic reactions) and that the presence of substrates and catalyst is sufficient for them to occur. Constraints on the endergonic reactions are explained below.

It is also important to remark that, in order to maintain a certain degree of generality, no hypotheses on the specific form of energy are formulated. Temperature is assumed to be kept constant by coupling the reactions with a heat bath.

Let us assume the presence of an incoming flux of loaded energy carriers  $\phi_E$  measured in ( $mol/sec$ ), which transport

<sup>2</sup>We could assume the condensations to be endergonic reactions and that, conversely, cleavage reactions occur spontaneously and do not require any chemical energy: these conditions hold, for example, in case of RNA in aqueous environment.

the energy into the system and which instantaneously diffuse in the reactor. The energy carriers, *ECs* from now on, bind and energize the internal species with a energization kinetic constant  $k_{nrg}$ .

Once that an energy carrier has released energy to a specific molecule it is removed from the system (we do not consider the unloaded energy carriers in the dynamics), while that species remains energized until it becomes part of any reactions requiring energy to proceed to completion. We also assume the presence of an outgoing flux of *ECs* coherent with the efflux constant of the reactor  $k_{out}$  and the presence of an *ECs* decay constant  $k_{dec}$ , by which an *EC* can be discarded because of the loss of its energetic load. Such processes are described as in the following:

$$\begin{cases} \frac{d[EC]}{dt} = \phi_E - k_{out}[EC] - k_{dec}[EC] \\ \quad - k_{nrg}[EC][X^-] \\ \frac{d[X^+]}{dt} = k_{nrg}[EC][X^-] - k_{out}[X^+] \\ \quad - k_{dec}[X^+] - \psi \\ \frac{d[X^-]}{dt} = \psi + K + k_{dec}[X^+] - k_{nrg}[EC][X^-] \\ \quad - k_{out}[X^-] \end{cases} \quad (2)$$

where  $[EC]$  stands for the concentration of the *ECs*,  $[X^+]$  represents the overall concentration of the charged molecules,  $[X^-]$  is the total concentration of the uncharged molecules,  $\psi$  represents the decrease of  $[X^+]$ , and the increase of  $[X^-]$ , because of the reactions occurred consuming the energy contained in the species involved in, and  $K$  represents the incoming flux (moles/sec) of uncharged molecules<sup>3</sup>.

It is important to stress that, considering the three-molecular nature of the condensation reactions, and the bi-molecular nature of the cleavage reactions, there are 12 possible combinatorial energy configurations in accordance with the position of the molecules carrying the energetic group: catalyst and/or first substrate and/or second substrate, table 1.

In accordance with table 1, the reactions admitted by the possible different energy frameworks can be thought as two independent Boolean functions, one for the condensation reactions and one for the cleavage reactions, of the respectively 8 and 4 possible input arguments (there are  $2^k$  possible different Boolean functions, where  $k$  is the number of different Boolean inputs).

Nevertheless, only a subset of the overall  $256 + 16$  Boolean functions are biologically plausible according to the adopted assumptions.

<sup>3</sup>Although the model is based on a stochastic algorithm, in order to speed up the computational performance, both the energy flux and the species energization processes are described by means of an approximated algorithm.

	Catalyst	Substrate 1	Substrate 2
Condensation			
1	+	+	+
2	+	+	-
3	+	-	+
4	+	-	-
5	-	+	+
6	-	+	-
7	-	-	+
8	-	-	-
Cleavage			
9	+	+	//
10	+	-	//
11	-	+	//
12	-	-	//

Table 1: In the table all the possible energy configurations are represented. Symbol “+” stands for the charged state of the molecules whereas symbol “-” represents the uncharged state of the molecule.

In principle, if we consider a system composed of a set of distinct interacting chemicals, it would be reasonable to assign distinct energetic Boolean functions to each specific reaction. Of course, there are constraints between cleavage and condensation groups (for a nice and detailed presentation, see (Plasson and Bersini, 2009)); at the present stage of the model we make simple choices compliant with the underlying physical and chemical properties (see below), by leaving more detailed and complex scenarios to future developments.

## Preliminary results

The preliminary analyses regarding the introduction of energy within the model are aimed to understand the influence of a variation of a) the energy carriers incoming flux  $\phi_E$  and of b) the energization kinetic constant  $k_{nrg}$  on the overall dynamics, with particular attention to the emergence of ACSs.

In detail, we considered the specific case in which all the condensations are *endoergonic* reactions and, thus, require energy to occur, and all the cleavages are neutral, since they can occur both in presence and in absence of energy. Furthermore, we decided to concentrate on the case in which a unique Boolean energy function is associated to all the condensation reactions, i.e. the function number 14 (00001110 in binary code), which requires that at least one of the two substrates is energized, while the catalyst of the reaction is necessarily not energized. For the sake of completeness, the Boolean energy function associated to the cleavage is the number 15 (1111 in binary code), that is the *true* function.

We specify that, in the course of this study, we decided

to simulate systems with standard structural parameters<sup>4</sup> and with a *critical* reaction probability, i.e. the probability according to which one random species catalyses, on the average, one random reaction<sup>5</sup>. We made this choice in order to observe whether and to what extent an energy variation in the system affects the emergence of ACSs in the region of the parameters space that is, according to the literature (Farmer and Kauffman, 1986), close to the phase transition.

We analyzed different ensembles of systems in which we varied independently:

- the incoming flux of energy carriers  $\phi_E$ , starting from the benchmark condition in which no carriers are present in the system: 0,  $10^{-23}$  mol/sec (corresponding to 6 carriers/sec),  $10^{-22}$  (60),  $10^{-21}$  (600),  $10^{-20}$  (6000);
- the energization kinetic constant  $k_{nrg}$ :  $10^{-1}$ , 1, 10,  $10^2$ ,  $10^3$ .

In figure 1 we can observe the variation of the average number of species present in the reactor, and not belonging to the incoming flux, at the end of the simulation (i.e. 1000 seconds) as a function of the variation of the incoming flux of energy carriers  $\phi_E$  (x axis) and of  $k_{nrg}$  (z axis).

In those cases in which there are no energy carriers in the system we can see that no new new species are present at the end of the simulation and this is clearly due to the impossibility for the condensations to occur in case of no energy. On the other hand, we can observe a maximum region along

<sup>4</sup>The detailed setting of the system is the following:

- the alphabet is composed of two letters, A and B;
- the firing disk containing the elements present in the reactor at the beginning of the simulation is composed of all the species up to length 4;
- the volume of the reactor is set to  $10^{-18}$  dm<sup>3</sup> and the overall initial concentration is set to  $10^{-4}$ ;
- the influx is composed of all the species of the firing disk and the influx rate is set to  $10^{-21}$  mol/sec;
- monomers and dimers can not be catalysts;
- the number of energy carriers entering the reactor and the value of the energy kinetic constant are varied according to the analyses and they are shown in the captions of the relative figures.

Notice that with these settings around 600 new molecules are entering the reactor every second and that at the theoretical dynamical equilibrium around 30000 molecules would be present inside the reactor.

<sup>5</sup>In this case the reaction probability is set to:  $10^{-4}$

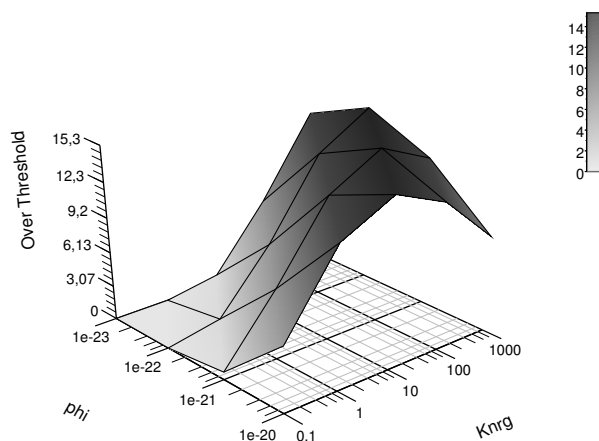


Figure 1: The figure shows the average number of species not belonging to the influx, with concentration greater than 0 from a set of 30 different simulations for each point represented in the graph. On the x axis the variation of  $\phi_E$  is represented while on the z axis the variation of  $k_{nrg}$  is represented.

the direction of the diagonal individuated by the following combination of  $\phi_E$  and  $k_{nrg}$  (respectively): ( $10^{-23}$  - 1;  $10^{-22}$  - 10;  $10^{-21}$  -  $10^2$ ;  $10^{-20}$  -  $10^3$ ), the maximum of the slope being reached in the cases corresponding to the following three combinations of  $\phi_E$  and  $k_{nrg}$  (respectively): ( $10^{-22}$  - 10;  $10^{-21}$  -  $10^2$ ;  $10^{-20}$  -  $10^3$ ), the first one being the absolute maximum.

Even if  $\phi_E$  and  $k_{nrg}$  are independent parameters, their combination actually represents the amount of available energy present in the system: this is the reason why similar values of the variable under analysis (i.e. the number of new species) are observed in relation to different combinations of these parameters. Moreover, the presence of a region of maximum indicates that there is an optimal amount of energy for the system in terms of overall production of new species. For larger values of both  $\phi_E$  and  $k_{nrg}$  the “efficiency” of the system in producing new species begins to decrease. This effect is partially due to the fact that when all the molecules in the reactor are energized the number of not-energized catalysts decrease due to the constraint on the total quantity of energy, as well as the number of possible condensations; hence in accordance with the particular assumptions concerning the chosen energy function, a decrement of the not-energized catalysts slows down the production of new molecules.

In figure 2 we can find the variation of the average concentration produced within the ACSs and within their first-order leaves in correspondence of the above mentioned combinations of  $\phi_E$  and  $k_{nrg}$ .

In figure 3 the variation of the average concentration of

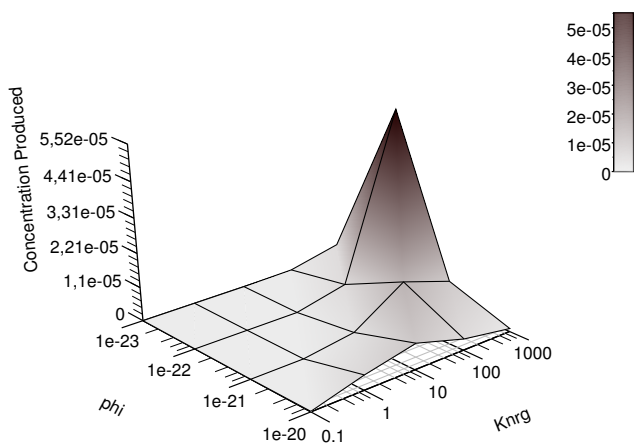


Figure 2: The figure shows the average concentration produced within the ACSs and within their first-order leaves from a set of 30 different simulations for each point represented in the graph. On the  $x$  axis the variation of  $\phi_E$  is represented while on the  $z$  axis the variation of  $k_{nrg}$  is represented.

the species produced by chains of reactions (and not belonging to ACSs) is shown.

We can see that, while the graph regarding the chains closely resembles that of the new species produced by the system, figure 1, confirming an optimal value of available energy in regard to the enhancement of the general activity, for what concerns the molecules produced within the ACSs (and their leaves) a unique point of maximum is reached for the combination  $\phi_E = 10^{-22}$  and  $k_{nrg} = 10^3$ , which also corresponds to the maximum in the creation of new species. Finally, it is important to remark that with most of the combinations of  $\phi$  and  $k_{nrg}$  no ACSs are present in the system at the end of the simulation and this would provide another possible explanation for the difficulty in observing the emergence of ACSs in wet lab experiments: according to these results, a fine tuning of the parameters regarding the energy is needed to allow the system to produce ACSs.

## Conclusions

The introduction of energy constraints associated to specific types of reactions represents a major novelty in the development of our stochastic model of catalytic reaction networks. In this regard, the main aim of this work was to show whether and to what extent the introduction of energy might affect the overall dynamics and, in particular, the emergence of autocatalytic cycles.

To this end, the preliminary analyses on critical systems showed that the combination of two key parameters, namely the incoming flux of energy carriers  $\phi$  and the energization

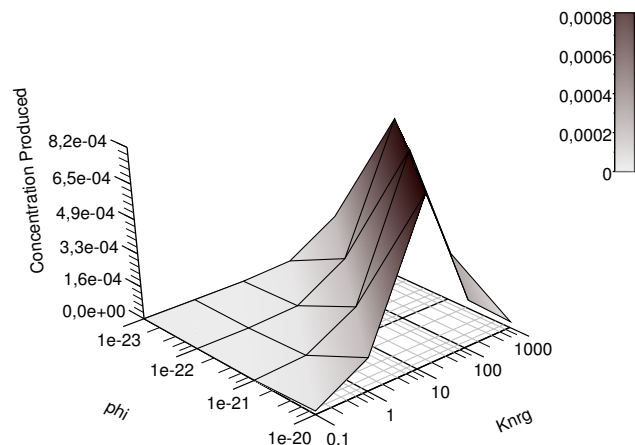


Figure 3: The figure shows the average concentration produced by chains of reactions from a set of 30 different simulations for each point represented in the graph. On the  $x$  axis the variation of  $\phi_E$  is represented while on the  $z$  axis the variation of  $k_{nrg}$  is represented.

kinetic constant  $k_{nrg}$ , jointly representing the amount of energy available for the endoergonic reactions, is responsible for a remarkable variation of the general activity of the system (indirectly attested by the production of new species). In particular, it was possible to prove the existence of an optimal value of energy, beyond which the activity of the system begins to decrease.

Focusing on the emergence of ACSs, it was then possible to demonstrate that the maximum production of new species is observed in the case of systems with optimal values of energy, which contain ACSs involving a large number of molecules, hence confirming their relevance in the overall dynamics. On the other hand, with most of the tested combinations of  $\phi$  and  $k_{nrg}$  ACSs could not be detected and this might provide another possible explanation to the difficulty in observing their emergence in wet lab experiments. Moreover, as we already showed in (Filisetti et al., 2011a, 2010), the autocatalytic sets are not robust and in most of them the catalytic closure is achieved by means of a “bottleneck” reaction occurring rarely during the simulations; although the energy constraints allow the emergence of structural ACSs, they do not confer neither robustness nor some forms of self-sustaining dynamics.

The results show that our model might unravel some unexpected features concerning the emergence of autocatalytic sets of molecules as for example the presence of an optimum in the energy flux.

Thus, several developments are underway in order to refine the description of the model like, for instance, the associa-

tion of distinct energization Boolean functions and of distinct  $k_{nrg}$  to different reactions and species, with the purpose of investigating possible complex behaviors related to the availability of energy.

## Acknowledgements

This work has been partially supported by the Fondazione di Venezia, <http://www.fondazionedivenezia.it>, (DICE project). Useful discussions with Stuart Kauffman, Norman Packard and Ruedi Fuchslin are gratefully acknowledged.

## References

- Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2005). *Molecular Biology of the Cell*, Fifth Edition. Garland Science, 4 edition.
- Anastasi, C., Buchet, F. F., Crowe, M. A., Parkes, A. L., Powner, M. W., Smith, J. M., and Sutherland, J. D. (2007). Rna: prebiotic product, or biotic invention? *Chemistry & biodiversity*, 4(4):721–39.
- Bagley, R. and Farmer, J. D. (1992). Spontaneous emergence of a metabolism. *Artificial Life II. Santa Fe Institute Studies in the Sciences of Complexity*, X:93–141.
- Bagley, R. J., Farmer, J. D., Kauffman, S. A., Packard, N. H., Perelson, A. S., and Stadnyk, I. M. (1989). Modeling adaptive biological systems. *Bio Systems*, 23(2-3):113–37; discussion 138.
- Bartel, D. P. (1999). *Re-creating an RNA Replicase*, chapter 5, pages 1431–144.
- Bartel, D. P. and Unrau, P. J. (1999). Constructing an rna world. *Trends in cell biology*, 9(12):M9–M13.
- Budin, I. and Szostak, J. W. (2010). Expanding roles for diverse physical phenomena during the origin of life. *Annual review of biophysics*, 39:245–63.
- De Lucrezia, D., Anella, F., and Chiarabelli, C. (2007). Question 5: on the chemical reality of the rna world. *Orig Life Evol Biosph*, 37(4-5):379–385.
- Dyson, F. J. (1985). *Origins of life*. Cambridge: Cambridge University Press.
- Eigen, M. and Schuster, P. (1977). The hypercycle. a principle of natural self-organization. part a: Emergence of the hypercycle. *Die Naturwissenschaften*, 64(11):541–65.
- Farmer, J. and Kauffman, S. (1986). Autocatalytic replication of polymers. *Physica D: Nonlinear Phenomena*, 220:50–67.
- Ferris, J. P., Hill, A. R., Liu, R., and Orgel, L. E. (1996). Synthesis of long prebiotic oligomers on mineral surfaces. *Nature*, 381(6577):59–61.
- Filiseti, A., Graudenzi, A., Serra, R., Villani, M., De Lucrezia, D., Fuchslin, R. M., Kauffman, S. A., Packard, N., and Poli, I. (2011a). A stochastic model of the emergence of autocatalytic cycles. *Journal of Systems Chemistry (in press)*.
- Filiseti, A., Serra, R., Villani, M., Fuchslin, R. M., Packard, N. H., Kauffman, S. A., and Poli, I. (2010). A stochastic model of autocatalytic reaction networks. In *Proceedings of the European Conference on Complex Systems (ECCS)*, Lisbon, September 13–17.
- Filiseti, A., Serra, R., Villani, M., Graudenzi, A., Fuchslin, R. M., and Poli, I. (2011b). The influence of the residence time on the dynamics of catalytic reaction networks. *Frontiers in Artificial Intelligence and Applications - Neural Nets WIRN10 - Proceedings of the 20th Italian Workshop on Neural Nets*, pages 243–251.
- Fishkis, M. (2007). Steps towards the formation of a protocell: the possible role of short peptides. *Origins of life and evolution of the biosphere : the journal of the International Society for the Study of the Origin of Life*, 37(6):537–53.
- Fuchslin, R. M., Filiseti, A., Serra, R., Villani, M., DeLucrezia, D., and Poli, I. (2010). Dynamical stability of autocatalytic sets. In Fellermann, H., Dörr, M., Hanczyc, M. M., Laursen, L. L., Maurer, S., Merkle, D., Monnard, P.-A., Stoy, K., and Rasmussen, S., editors, *Artificial Life XII, Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*, pages 65–72. The MIT Press.
- Gilbert, W. (1986). Origin of life: The rna world. *Nature*, 319(6055):618–618.
- Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361.
- Gillespie, D. T. (2007). Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry*, 58(1):35–55.
- Jain, S. and Krishna, S. (1998). Autocatalytic set and the growth of complexity in an evolutionary model. *Phys Rev Lett*, 81:5684–5687.
- Kaneko, K. (2006). *Life: An Introduction to Complex Systems Biology (Understanding Complex Systems)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- Kauffman, S. A. (1986). Autocatalytic sets of proteins. *J Theor Biol*, 119(1):1–24.
- Kauffman, S. A. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, USA, 1 edition.
- Lee, D. H., Granja, J. R., Martinez, J. A., Severin, K., and Ghadri, M. R. (1996). A self-replicating peptide. *Nature*, 382(6591):525–8.
- Lifson, S. (1997). On the crucial stages in the origin of animate matter. *Journal of molecular evolution*, 44(1):1–8.
- Ma, W., Yu, C., Zhang, W., Zhou, P., and Hu, J. (2010). Self-replication: spelling it out in a chemical background. *Theory in biosciences*.
- Miller, S. L. (1953). A production of amino acids under possible primitive earth conditions. *Science*, 117:528–529.
- Mossel, E. and Steel, M. (2005). Random biochemical networks: the probability of self-sustaining autocatalysis. *Journal of theoretical biology*, 233(3):327–36.

- Müller, U. F. (2006). Re-creating an rna world. *Cellular and molecular life sciences : CMLS*, 63(11):1278–93.
- Oparin, A. (1924). *The origin of life on the earth*. Oliver and Boyd, 1st ed., p edition.
- Plasson, R. and Bersini, H. (2009). Energetic and entropic analysis of mirror symmetry breaking processes in a recycled microreversible chemical system. *The journal of physical chemistry. B*, 113(11):3477–90.
- Rios, A. C. and Tor, Y. (2009). Model systems: how chemical biologists study rna. *Current opinion in chemical biology*, 13(5-6):660–8.
- Ruiz-Mirazo, K. and Mavelli, F. (2008). On the way towards 'basic autonomous agents': stochastic simulations of minimal lipid-peptide cells. *Bio Systems*, 91(2):374–87.
- Saghatelian, A., Yokobayashi, Y., Soltani, K., and Ghadiri, M. R. (2001). A chiroselective peptide replicator. *Nature*, 409(6822):797–801.
- Segre, D., Lancet, D., Kedom, O., and Pilpel, Y. (1998). Graded autocatalysis replication domain (gard): kinetic analysis of self-replication in mutually catalytic sets. *Orig Life Evol Biosph*, 28(4-6):501–514.
- Talini, G., Gallori, E., and Maurel, M.-C. (2009). Natural and unnatural ribozymes: back to the primordial rna world. *Research in microbiology*, 160(7):457–65.

# Distinguishing the effects of epistasis and pleiotropy using a variant of the NK model

Jonathan M. Fisher<sup>1</sup> and Jason H. Moore<sup>1</sup>

<sup>1</sup>Dartmouth Medical School, One Medical Center Drive, HB 7937, Lebanon, NH 03766  
jonathan.m.fisher@dartmouth.edu

## Abstract

Pleiotropy and epistasis are central to understanding how genes are expressed. Kauffman's NK model is used ubiquitously to investigate gene expression in evolution and other contexts; it is widely understood to reflect the results of epistasis, but it is less often used to study pleiotropy. In this paper we introduce the NEP model, a variant of the NK model which allows epistasis and pleiotropy to be studied individually. We apply our methods to global and local optima and adaptive walks, and elucidate new insights into Kauffman's complexity catastrophe.

## Introduction

Pleiotropy, which refers to a single locus affecting more than one trait, and epistasis, several loci collectively affecting a single trait, have long been recognized to be fundamental to our understanding of gene expression (Tyler et al., 2009). Epistasis is widely encountered in humans (Moore, 2003) and other organisms (Remold and Lenski, 2004; Bonhoeffer et al., 2004), as is pleiotropy (Ostrowski et al., 2005; Wagner et al., 2008; Scarcelli et al., 2007). Epistasis and pleiotropy are also seen to play a key role in evolution (Phillips, 2008; Fenster et al., 1997). Thus it is important to form a clear picture of the mechanisms of epistasis and pleiotropy and their effects on phenotypes.

Kauffman's NK model (Kauffman and Levin, 1987), a computational model of genomes in fitness landscapes, has been widely used to investigate properties of fitness spaces (Kauffman, 1993; Weinberger and Stadler, 1993; Macken and Perelson, 1989; Orr, 2005) and evolution thereon (Østman et al., 2010). A number of variants of the NK model have also been studied, such as the infinite-allele variant (Welch and Waxman, 2005) and the block model (Perelson and Macken, 1995); the NK model and its variants have been shown to be applicable to a variety of biological phenomena (Macken and Perelson, 1989; Perelson and Macken, 1995; Kauffman and Weinberger, 1989; Orr, 2006).

Epistasis and pleiotropy can be tuned in the NK model, but they always vary in tandem, which makes it difficult to study the two effects separately. In this paper we describe

the NEP model, a variant of the NK model in which epistasis and pleiotropy can be tuned independently.

## Methods

### Models

**The NK Model** The NK model comprises a population of genomes, each of which consists of  $N$  loci, with  $A$  alleles at each locus. The model defines one trait for each locus; the locus interacts epistatically with  $K$  other loci in determining that trait. The fitness of a genome is calculated by averaging the fitness contributions of all of the traits. Each trait is represented by a  $(K + 1)$ -dimensional table, with the length along each dimension equal to  $A$ , where the values in the table are stochastically chosen from a uniform distribution. The fitness contribution for each trait is selected from the table by choosing the row corresponding to the allele of the base locus, the column corresponding to the allele of the next locus, etc.

Choosing which other loci interact with a given locus can either be done deterministically, by having each locus interact with the succeeding  $K$  loci (where the genome is assumed to be circular), or stochastically, by choosing  $K$  other loci at random. Because the NK model contains a trait/table for each of the  $N$  loci, there are  $N$  traits/tables in total. In the rest of the paper we will refer to traits and tables interchangeably.

Fig. 1 gives an example genome with its associated tables; the top part of this figure refers to the NK model, and the bottom part contains tables that are added for the NEP model. In this example  $N = 4$  and  $A = 2$ , so each genome contains 4 loci with 2 alleles each; the interaction degree,  $K$ , is 1. The horizontal bar in the middle of the diagram is the genome, and the 4 dots on the bar are the loci. The numbers 0, 1, 1, 0 along the genome are the alleles at each locus.

In this example, each consecutive pair of loci interact in a trait, as do the outer two loci, for a total of four traits. Each trait is shown in the diagram as a two-dimensional lookup table above the genome, which is linked to its pair of loci by a pair of lines. The first pair of alleles is (0, 1), so the corresponding fitness contribution is the .71 shown in the



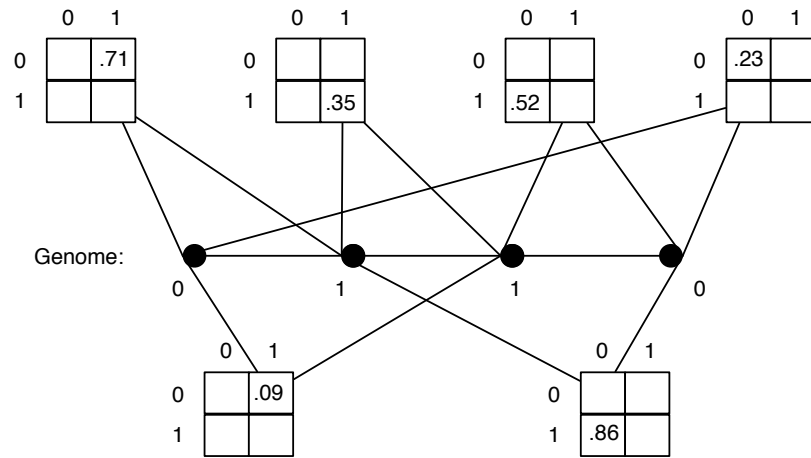


Figure 1: An example genome with its tables. The genome, with alleles, runs across the center of the figure; each pair of loci is linked to a table which represents the trait determined by that pair of loci. The single entry shown in each table is the fitness chosen by the allele values of the corresponding loci. The four tables along the top of the figure constitute an NK model, and all six tables together form an NEP model.

0th row and the 1st column; the fitness contributions for the other pairs of loci are shown similarly. (In an actual instance of the model, of course, all of the values in each table would be filled in.) The overall fitness of the genome is then the average of the four lookup values.

The parameter  $K$  has traditionally been used to tune the degree of epistasis in the model:  $K$  determines the degree of epistasis, because each locus in the model interacts with  $K$  other loci. However, we note that  $K$  also determines the degree of pleiotropy, because each locus appears in  $K + 1$  tables. The NK model has no way to separate epistasis from pleiotropy, however, which can lead to uncertainty about which of the two is causing any particular observed effect.

**The NEP Model** In order to separate epistasis from pleiotropy, we introduce the NEP model, adding the two new parameters  $E$  and  $P$ . To assist in describing  $E$  and  $P$ , we define  $T$  as the number of tables in the model. For  $i$  from 1 to  $T$ ,  $E_i$  is the number of loci in the  $i$ th table — in other words, the number of loci that are used to look up the value in the  $i$ th table. The loci used in each table are chosen at random. We refer to  $E_i$  as the epistatic dimension of table  $i$ . For  $j$  from 1 to  $N$ ,  $P_j$  is the number of tables in which the  $j$ th locus appears, referred to as the pleiotropic dimension of that locus. The NEP model is a generalization of the NK model, and the parameters  $E$  and  $P$  can be seen in the NK model: since the parameter  $K$  refers to how many *other* loci a given locus interacts with, while our parameter  $E$  refers to the *total* number of interacting loci, any given NK model has  $K = E - 1$ ; similarly,  $K = P - 1$ . The difference between the NK model and the NEP model is that the NK model fixes  $P \equiv E$  but the NEP model allows  $P$  to be different from  $E$ .

When two or more loci interact epistatically in the NEP

model, the fitness contribution for that interaction is chosen from a stochastic lookup table. The stochasticity of the lookup table means that the interaction between the loci in determining the fitness is highly non-linear, consistent with the usual definition of epistasis.

Since a genome can have  $A$  possible values at each locus, there are a total of  $A^N$  genomes in the fitness space in either the NK model or the NEP model.

In the example of the NK model in Fig. 1 described above,  $E = 2$  and  $P = 2$ ; as always in the NK model,  $P \equiv E$ . However, in the NEP model we can modify the example to have  $P$  differ from  $E$ , by adding the two tables shown below the genome. The dimension  $E$  of each table is still 2, but now each locus is linked to 3 tables, so  $P = 3$ .

Referring to Fig. 1, there are two ways to count the links between the genome and the trait tables: If we count the links where they connect to the tables, we get  $\sum_{i=1}^T E_i$  since the epistatic dimension of each table is equal to the number of links connecting to it. On the other hand, if we count the links where they connect to the loci on the genome, we get  $\sum_{j=1}^N P_j$  since the pleiotropic dimension of each locus is equal to the number of links connecting to it. Since those two counts must be equal, we find that

$$\sum_{j=1}^N P_j = \sum_{i=1}^T E_i. \quad (1)$$

This equation can be rewritten as  $N\bar{P} = \bar{E}T$ , where  $\bar{P}$  and  $\bar{E}$  are the average values of  $P_j$  and  $E_i$  respectively. In practice we usually either choose to have  $P_j \equiv P$ , or choose to have  $E_i \equiv E$ . In this paper we take the former approach.

Note that the embedded-landscape model (Altenberg,

1994; Heckendorn et al., 1999) has previously introduced flexibility in the number of tables to the NK model, in the same way as the NEP model; however, the embedded-landscape model has generally been used in studies of computational complexity and has typically not delineated the effects of epistasis and pleiotropy.

## Numerical Analysis

The current study employs an NEP model with  $N = 20$  and  $A = 2$ . We simulated 400 different models: for  $P$  equal to each of 1 through 20 we chose, using the equation  $NP = \bar{E}T$ , the value of  $\bar{E}$  closest to each of 1 through 20, for a total of 400  $(P, \bar{E})$  pairs. When  $\bar{E}$  was an integer we chose all of the tables to have that value for  $E_i$ . When  $\bar{E}$  was not an integer we chose some of the tables to have  $E_i$  equal to the next integer below  $\bar{E}$ , and some the next integer above, with the counts of each chosen to give the desired average  $\bar{E}$ . This ensured that the tables were as similar in size as possible. Each pair  $(P, \bar{E})$  gave a model, and each model was run 100 times, populating the lookup tables with a different random seed each time; all of the statistics for each model were averaged across those runs.

Among the phenomena we study here are local optima and adaptive walks, both of which are important in studying the dynamics of evolution (Orr, 2005). If the fitness of a given genome is greater than the fitness of any genome at a Hamming distance of 1 from the given genome, that fitness is said to be a local optimum. An adaptive walk is a sequence of genomes that starts at a randomly-selected genome and proceeds by single fitness-improving mutations until it reaches a local optimum.

We collected the following statistics: variance in fitness within each fitness space; global optimum fitness, defined as the largest fitness value in the fitness space; mean local optimum fitness; number of local optima in the fitness space; average length of adaptive walks, defined as the number of fitness-improving mutations traversed in the walk; fitnesses attained in adaptive walks, defined as the fitness of the last genome in the adaptive walk (by definition, a local optimum); first step up in adaptive walks, defined as the difference in fitness between the first genome and the second genome in an adaptive walk; last step up in adaptive walks, defined as the difference in fitness between the second-to-last genome and the last genome in an adaptive walk; and the maximum step up, defined as the largest difference in fitness between any two adjacent genomes in the fitness space. These are some of the most commonly-measured statistics in studies of the NK model (Kauffman, 1993; Macken and Perelson, 1989) and its variants (Perelson and Macken, 1995).

In each run, the variance, the global optimum, the mean local optimum, and the number of local optima were calculated by iterating across all  $2^{20}$  genomes in the fitness space. The maximum step up was calculated by iterating

across all pairs of genomes. To calculate statistics on adaptive walks, 1,000 adaptive walks were simulated in each run; each reported statistic is then the average value of that statistic across the 1,000 adaptive walks simulated.

Representative results from the 400 models are described below.

## Results

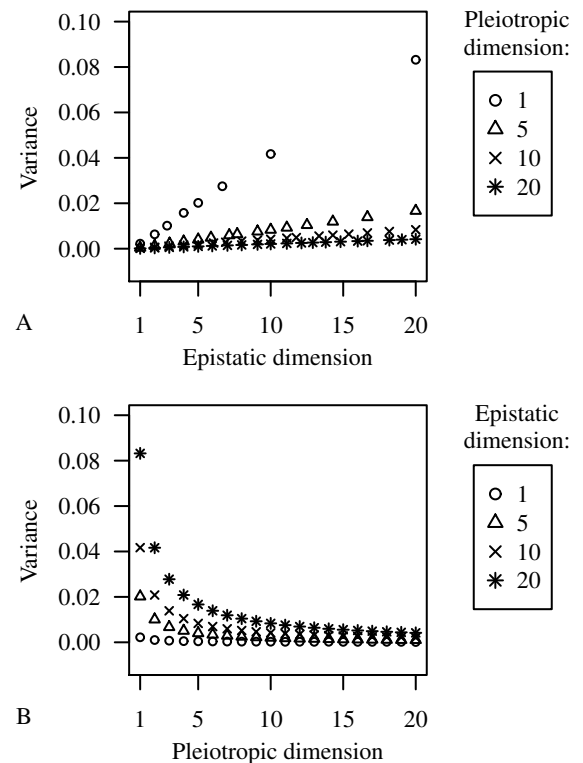


Figure 2: Fitness variance by epistatic dimension and pleiotropic dimension.

## Fitnesses and Optima

Variance in fitness increased approximately linearly with epistatic dimension (Fig. 2A) and decreased with pleiotropic dimension (Fig. 2B). This can be understood by considering what happens when we increment either the epistatic dimension or the pleiotropic dimension. First, if we increase  $\bar{P}$  by 1 while holding  $\bar{E}$  fixed, the equation  $N\bar{P} = \bar{E}T$  tells us that we need to multiply  $T$  by  $(\bar{P} + 1)/\bar{P}$ . Since the fitness of any given genome is calculated by averaging one entry from each of the  $T$  tables, multiplying  $T$  by  $(\bar{P} + 1)/\bar{P}$  means that each calculated fitness is averaged across more table-entries, which decreases the fitness variance.

Secondly, consider incrementing  $\bar{E}$  while holding  $\bar{P}$  fixed. This requires us to multiply  $T$  by  $\bar{E}/(\bar{E} + 1)$ , decreasing the value of  $T$ , which has the opposite effect to

incrementing  $\bar{P}$ : it increases the variance of fitnesses. Incrementing  $\bar{E}$  also increases the variance by increasing the degrees of freedom of the model, which is equal to the number of separately-generated stochastic table-entries. Since each table contains  $A^{\bar{E}}$  entries, for any given values of  $A$ ,  $\bar{E}$ , and  $T$ , the number of separately-generated stochastic table-entries is

$$TA^{\bar{E}}. \quad (2)$$

If we increment  $\bar{E}$  while holding  $\bar{P}$  fixed then the new total number of entries is given by

$$T \frac{\bar{E}}{\bar{E} + 1} A^{\bar{E}+1}. \quad (3)$$

Thus the net effect of incrementing  $\bar{E}$  is to multiply the number of separately-generated stochastic table-entries, and thus the degrees of freedom, by  $A\bar{E}/(\bar{E} + 1)$ . For  $A = 2$ , this is greater than 1 as long as  $E > 1$ .

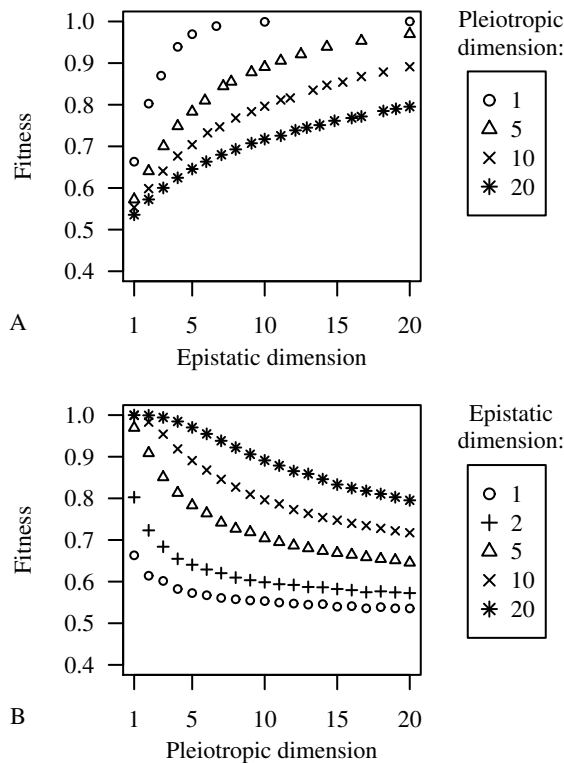


Figure 3: Global optimum fitness by epistatic dimension and pleiotropic dimension.

Fitnesses of global optima increased with epistatic dimension; for smaller pleiotropic dimension (up through about 5) the global optimum approached 1.0 (Fig. 3A). The global optima decreased with pleiotropic dimension (Fig. 3B). The global optimum is an extreme value of fitness, so as the variance increases with epistatic dimension, the global optimum

also increases. The opposite argument shows why the global optimum decreases with pleiotropic dimension.

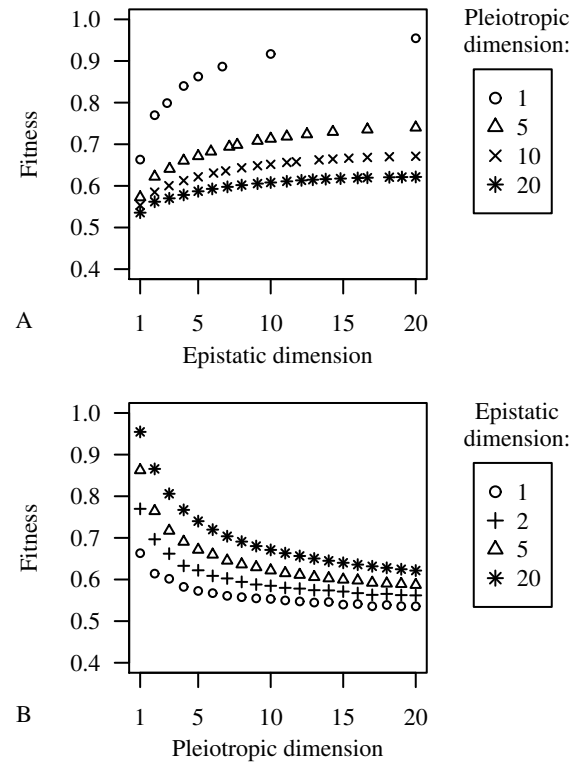


Figure 4: Local optimum fitness by epistatic dimension and pleiotropic dimension.

As with global optima, fitnesses of local optima increased with epistatic dimension (Fig. 4A) and decreased with pleiotropic dimension (Fig. 4B). Since local optima are local extreme values, the same argument as with global optima indicates why local optima increase with epistatic dimension and decrease with pleiotropic dimension.

The number of local optima increased dramatically with epistatic dimension (Fig. 5A); for high and low epistatic dimensions, the number of local optima did not vary with pleiotropic dimension, but for intermediate epistatic dimensions, lower pleiotropic dimensions gave slightly higher numbers of local optima (Fig. 5B). Note also that for an epistatic dimension of 2, the graph of the number of local optima by pleiotropic dimension is a little noisy; later we will notice the same phenomenon in a different way with the lengths of adaptive walks.

Kauffman proves (Kauffman, 1993) that for  $K = N - 1$  the number of local optima is very large; the proof uses only the epistasis effects of  $K$ , not the pleiotropy effects. The results we get here are consistent with that, in that the number of local optima increases with epistatic dimension but varies little with pleiotropic dimension.

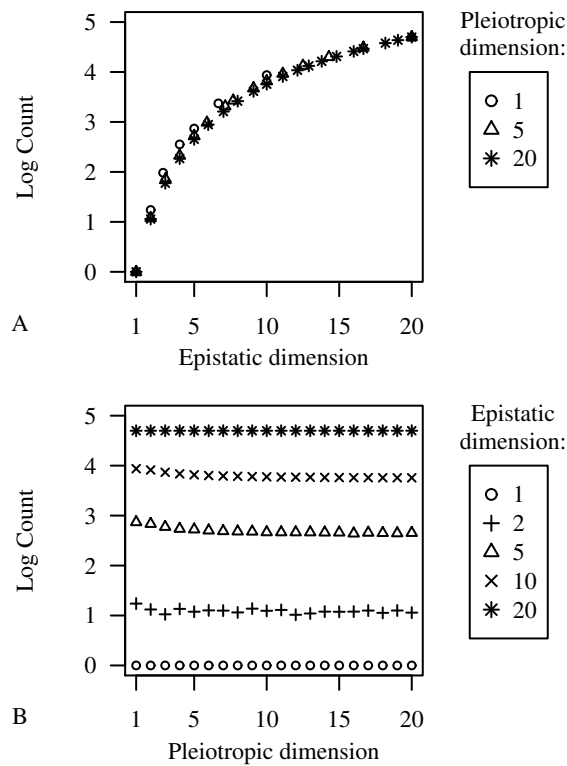


Figure 5: Log number of local optima by epistatic dimension and pleiotropic dimension.

## Adaptive Walks

In general the length of adaptive walks decreased with epistatic dimension (Fig. 6A) and increased or held steady with pleiotropic dimension (Fig. 6B). For epistatic dimension of 1 the average length was 10, regardless of pleiotropic dimension; otherwise, for each pleiotropic dimension, the average length peaked at an epistatic dimension of 2 and decreased from there. For epistatic dimension between 2 and about 10, lengths of adaptive walks increased as a function of pleiotropic dimension.

Kauffman proves (Kauffman, 1993) that in an NK model where  $A = 2$  and  $K = 0$  the average length of an adaptive walk is  $N/2$ . Again, the proof uses only the epistasis effects of  $K$ , and thus holds for the NEP model; this explains why the average length is 10 in our models when the epistatic dimension is 1 (which is equivalent to  $K = 0$ ). For epistatic dimensions greater than 1, the average adaptive-walk length is inversely related to the number of local optima, and thus it decreases with epistatic dimension and increases with pleiotropic dimension for low epistatic dimensions. The inverse relationship with number of local optima is also why we see the same noisiness in the graph of adaptive-walk length by pleiotropic dimension for epistatic dimension of 2 as we did with the number of local optima.

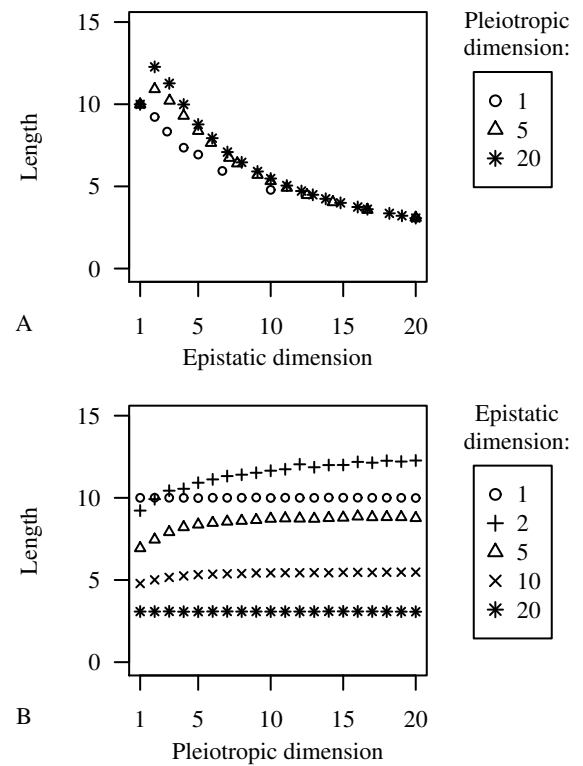


Figure 6: Length of adaptive walks by epistatic dimension and pleiotropic dimension.

Attained fitness in adaptive walks increased with epistatic dimension, with the effect being most pronounced for low pleiotropic dimension (Fig. 7A); attained fitness decreased with pleiotropic dimension (Fig. 7B).

Adaptive-walk lengths decrease with epistatic dimension but attained fitnesses increase; the opposite is true with respect to pleiotropic dimension. This apparent contradiction is resolved by Figures 8 and 9, which show the typical steps in adaptive walks.

The typical first step up in an adaptive walk increased with epistatic dimension (Fig. 8A) and decreased with pleiotropic dimension (Fig. 8B). The same is true for the typical last step (Fig. 9), with one exception: for epistatic dimensions greater than about 10, the typical last step up was lower for a pleiotropic dimension of 1 than for the next few pleiotropic dimensions. Again we see the effects of variance: increasing the variance with epistatic dimension allows for larger steps up, while decreasing it with pleiotropic dimension decreases the possible steps up. The phenomenon whereby average steps up increase with increasing  $K$  in the NK model, allowing higher fitness to be achieved with fewer steps, has been observed by Østman et al. (2010), who ascribed the phenomenon to a combination of pleiotropy and epistasis. In contrast, we find that the increasing amplitude of steps up is due solely to increasing epistasis, not increasing pleiotropy,

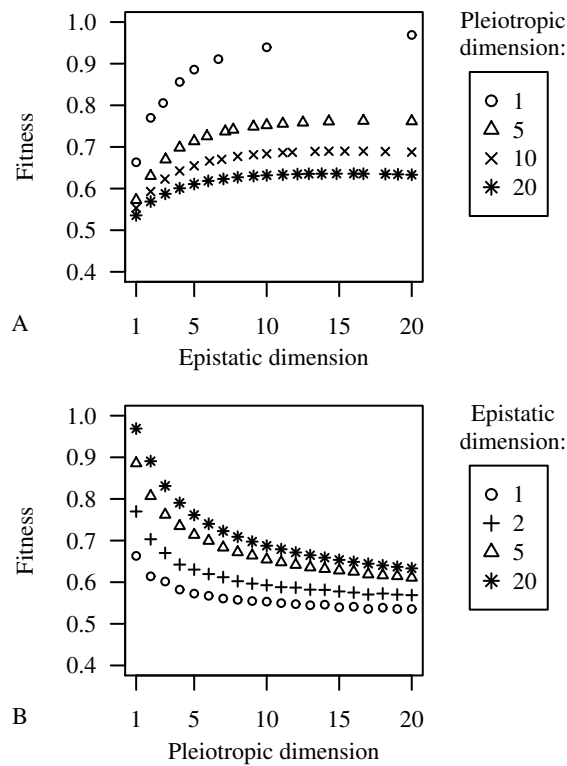


Figure 7: Attained fitness of adaptive walks by epistatic dimension and pleiotropic dimension.

and, in fact, that amplitudes of steps up decrease with increasing pleiotropy.

Except for an epistatic dimension of 1 (where they were the same), the typical last step was less than the typical first step for all epistatic dimensions and pleiotropic dimensions; the effect was more pronounced for higher epistatic dimension and lower pleiotropic dimension. The reason for this is that on the last step of an adaptive walk the fitness of the genome is already quite high, limiting the remaining steps available.

The maximum step up, defined as the largest difference in fitness between two adjacent genomes, increased with epistatic dimension (Fig. 10A) and decreased with pleiotropic dimension (Fig. 10B). Again we see the result of the fact that fitness variance increases with epistatic dimension and decreases with pleiotropic dimension.

## Discussion

In the NEP model, referring to the equation  $N\bar{P} = \bar{E}T$  we see that one way to increase the pleiotropic dimension  $P$  is by increasing  $T$  by adding one or more traits. As the results here have shown, increasing pleiotropy in this way tends to decrease the local and global optima. On the other hand, when a biological species adds a trait its fitness generally goes up; it may be that when a biological species adds a trait

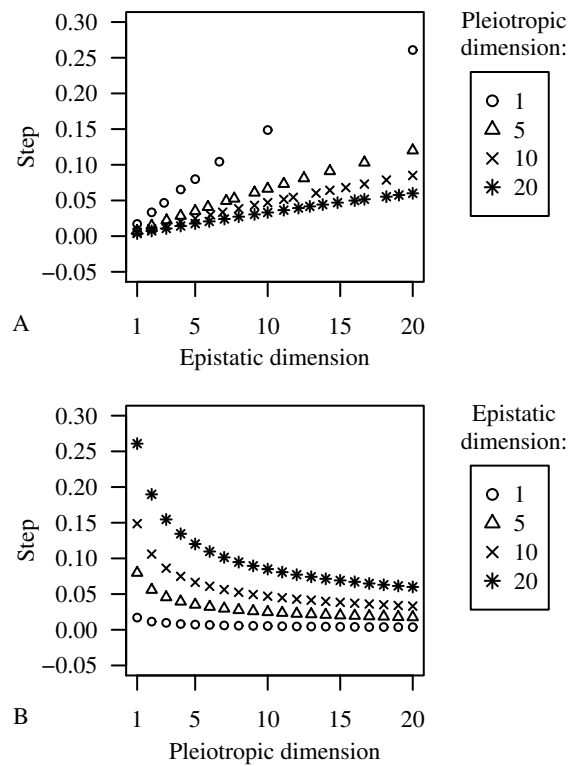


Figure 8: First step of adaptive walks by epistatic dimension and pleiotropic dimension.

it also increases the overall number of loci, thus avoiding an increase in pleiotropy. Further investigation will be required to resolve this question.

## Effects of Epistasis and Pleiotropy

The overall effect of epistasis is to increase the variance in fitness, and of pleiotropy to decrease it. We see this first in the direct measurements of variance of fitness; we also see it in measurements of global and local optima: less variance reduces the heights of the available maxima. Attained fitnesses in adaptive walks match closely with mean local optima, so they, too, decrease with decreasing variance. First and last steps in adaptive walks, and maximum single steps, also trend in the same direction as variance. The situation here is more complicated, however: because a single step means mutating a single allele, we are very far from the extreme-value considerations that apply to global and local optima.

A key difference between the NEP model and the embedded-landscape model is that the latter does not fully separate the effects of epistasis and pleiotropy. For example, (Smith and Smith, 1999) find that “the epistasis parameter  $K$  has little effect on the global fitness statistics, and does not affect the mean fitness values of the local optima at all.” However, when we separated epistasis out from pleiotropy,

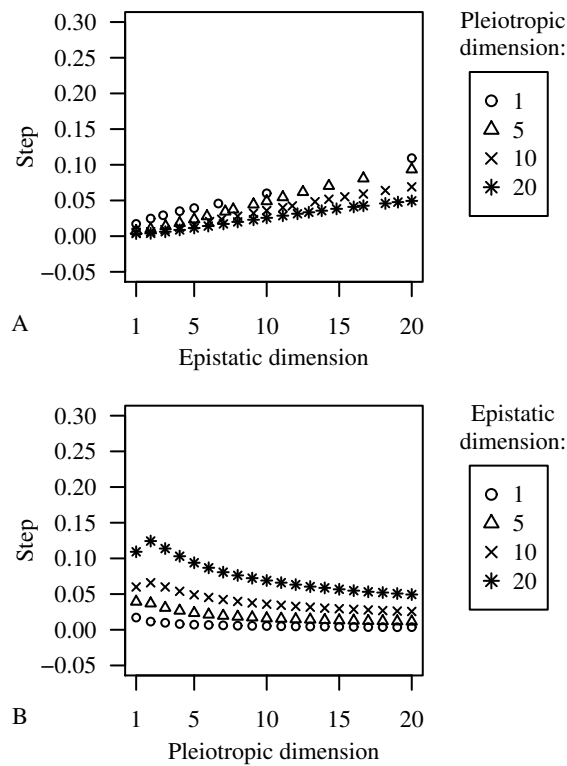


Figure 9: Last step of adaptive walks by epistatic dimension and pleiotropic dimension.

we found a clear dependency of global optimum fitness and mean local optimum fitness on epistasis.

### The Complexity Catastrophe

In working with the NK model, Kauffman observed a tendency for fitnesses of local optima to decrease with increasing  $K$ , and coined the term “complexity catastrophe” to describe this (Kauffman, 1993). Previous authors have ascribed the decrease in local optima to an increase in epistasis (e.g., Kauffman (1993); Solow et al. (1999)); in contrast, in the context of the embedded-landscape model (Smith and Smith, 1999) stated that the decrease in local optima is due to an increase in the number of traits.

The results described here allow us to clarify which part of the complexity causes the catastrophe. As pleiotropic dimension increases, both attained fitnesses and the fitnesses of local optima decrease. On the other hand, as epistatic dimension increases, both attained fitnesses and the fitnesses of local optima *increase*. The culprit in the complexity catastrophe is simply that decreasing variance lowers the local optima.

Kauffman further observed that as  $K$  increases, the mean value of local optima initially increases, and then starts to decrease. Looking at parts A and B of Fig. 4, we see the explanation of the observed trends: Increasing  $K$  corresponds

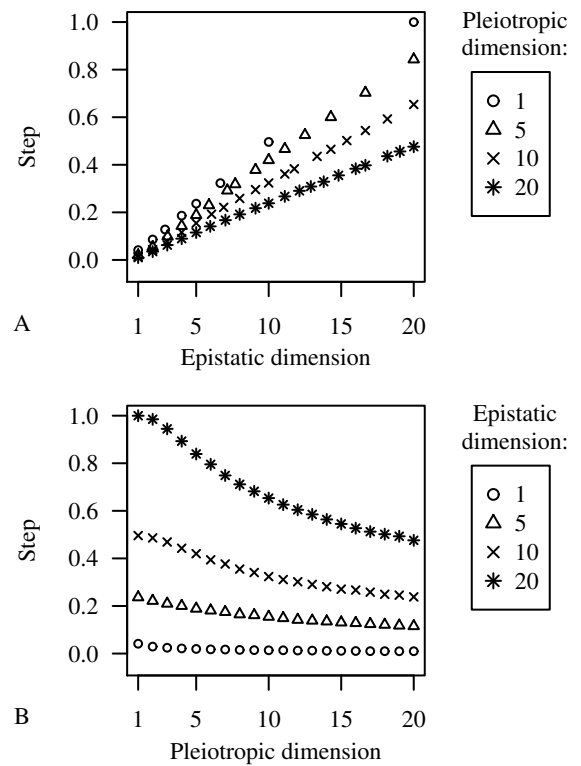


Figure 10: Maximum single step by epistatic dimension and pleiotropic dimension.

to increasing  $E$  and  $P$  simultaneously, which results in simultaneous tendencies to increase and decrease local optima, respectively; at first the former tendency predominates, and then the latter.

### Conclusion

Given the ubiquity of epistasis and pleiotropy in gene expression, and given the prevalence of the NK model in studying genetic phenomena, it is important to ensure that we fully understand those two forms of gene linkage in the context of the NK model. As described here, the NEP model can be used to distinguish the effects of epistasis and pleiotropy, allowing assumptions made in the literature to be re-examined, and allowing new insights to be gained going forward.

### Acknowledgements

We thank Davnah Urbach and Jeff Kiralis and our other colleagues in the lab for their assistance with preparing this paper. This work was supported by NIH grants LM009012, LM010098, and AI59694.

### References

Altenberg, L. (1994). Evolving better representations through selective genome growth. In Schaffer, J. D., Schwefel, H. P.,

- and Kitano, H., editors, *Proceedings of the IEEE World Congress on Computational Intelligence*, pages 182–187. IEEE, Piscataway, NJ.
- Bonhoeffer, S., Chappey, C., Parkin, N. T., Whitcomb, J. M., and Petropoulos, C. J. (2004). Evidence for positive epistasis in HIV-1. *Science*, 306:1547–1550.
- Fenster, C. B., Galloway, L. F., and Chao, L. (1997). Epistasis and its consequences for the evolution of natural populations. *Trends in Ecology & Evolution*, 12:282–286.
- Heckendorn, R. B., Rana, S., and Whitley, D. (1999). Test function generators as embedded landscapes. In Bäck, T. and Banzhaf, W., editors, *Foundations of Genetic Algorithms - 5*, pages 183–198. Morgan Kaufmann, San Francisco, CA.
- Kauffman, S. A. (1993). *The Origins of Order*. Oxford University Press, New York.
- Kauffman, S. A. and Levin, S. (1987). Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, 128:11–45.
- Kauffman, S. A. and Weinberger, E. D. (1989). The NK model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of Theoretical Biology*, 141:211–245.
- Macken, C. A. and Perelson, A. S. (1989). Protein evolution on rugged landscapes. *Proceedings of the National Academy of Sciences*, 86:6191–6195.
- Moore, J. H. (2003). The ubiquitous nature of epistasis in determining susceptibility to common human diseases. *Human Heredity*, 56:73–82.
- Orr, H. A. (2005). The genetic theory of adaptation: a brief history. *Nature Reviews Genetics*, 6:119–127.
- Orr, H. A. (2006). The population genetics of adaptation on correlated fitness landscapes: The block model. *Evolution*, 60:1113–1124.
- Østman, B., Hintze, A., and Adami, C. (2010). Impact of epistasis and pleiotropy on evolutionary adaptation. *arXiv:0909.3506v2*.
- Ostrowski, E. A., Rozen, D. E., and Lenski, R. E. (2005). Pleiotropic effects of beneficial mutations in *escherichia coli* evolution. *Evolution*, 59:2343–2352.
- Perelson, A. S. and Macken, C. A. (1995). Protein evolution on partially correlated landscapes. *Proceedings of the National Academy of Sciences*, 92:9657–9661.
- Phillips, P. C. (2008). Epistasis - the essential role of gene interactions in the structure and evolution of genetic systems. *Nature Reviews Genetics*, 9:855–867.
- Remold, S. K. and Lenski, R. E. (2004). Pervasive joint influence of epistasis and plasticity on mutational effects in *escherichia coli*. *Nature Genetics*, 36:423–426.
- Scarcelli, N., Cheverud, J. M., Schaal, B. A., and Kover, P. X. (2007). Antagonistic pleiotropic effects reduce the potential adaptive value of the FRIGIDA locus. *Proceedings of the National Academy of Sciences*, 104:16986–16991.
- Smith, R. E. and Smith, J. E. (1999). An examination of tunable, random search landscapes. In Bäck, T. and Banzhaf, W., editors, *Foundations of Genetic Algorithms - 5*, pages 165–182. Morgan Kaufmann, San Francisco, CA.
- Solow, D., Burnetas, A., Roeder, T., and Greenspan, N. S. (1999). Evolutionary consequences of selected locus-specific variations in epistasis and fitness contribution in Kauffman’s NK model. *Journal of Theoretical Biology*, 196:181–196.
- Tyler, A. L., Asselbergs, F. W., Williams, S. M., and Moore, J. H. (2009). Shadows of complexity: what biological networks reveal about epistasis and pleiotropy. *BioEssays*, 31:220–227.
- Wagner, G. P., Kenney-Hunt, J. P., Pavlicev, M., Peck, J. R., Waxman, D., and Cheverud, J. M. (2008). Pleiotropic scaling of gene effects and the ‘cost of complexity’. *Nature Letters*, 452:470–473.
- Weinberger, E. D. and Stadler, P. F. (1993). Why some fitness landscapes are fractal. *Journal of Theoretical Biology*, 163:255–275.
- Welch, J. J. and Waxman, D. (2005). The *nk* model and population genetics. *Journal of Theoretical Biology*, 234:329–340.



# A complex systems approach to education in Switzerland

Regina Frei<sup>1</sup>

<sup>1</sup>Intelligent Systems & Networks Group, Imperial College London, South Kensington, London SW7 2BT, UK  
work@reginafrei.ch

## Abstract

The insights gained from the study of complex systems in biological, social, and engineered systems enable us not only to observe and understand, but also to actively design systems which will be capable of successfully coping with complex and dynamically changing situations. The methods and mindset required for this approach have been applied to educational systems with their diverse levels of scale and complexity. Based on the general case made by Yaneer Bar-Yam, this paper applies the complex systems approach to the educational system in Switzerland. It confirms that the complex systems approach is valid. Indeed, many recommendations made for the general case have already been implemented in the Swiss education system. To address existing problems and difficulties, further steps are recommended. This paper contributes to the further establishment complex systems approach by shedding light on an area which concerns us all, which is a frequent topic of discussion and dispute among politicians and the public, where billions of dollars have been spent without achieving the desired results, and where it is difficult to directly derive consequences from actions taken. The analysis of the education system's different levels, their complexity and scale will clarify how such a dynamic system should be approached, and how it can be guided towards the desired performance.

## Introduction

The principles of complex systems have been successfully applied to a diversity of problems Bar-Yam (2004), including the health system, military warfare, international development and educational systems. Although still in its initial phase, the effects obtained through using a complex systems approach – also called “Enlightened Evolutionary Engineering” – are generally promising. At the moment, further validation to increase the credibility of this approach is still required. This paper thus applies the complex systems approach to the educational system in Switzerland. As it turns out, the Swiss system – which functions rather well in comparison with other education systems – already uses several of the recommended principles. Examples include offering a variety of ways towards professional qualifications, and using a diversity of actions to provide for the individual students' needs. To address remaining or new problems, the

complex systems approach should be applied consequently, as detailed in this article.

Complex systems can help us improve our educational systems by making us understand the differences between diverse levels within the educational system and the respective approaches they require. We will understand why certain educational systems perform better than others by discussing the “one fits all” uniform large-scale approach as compared to diversity and individuality.

The motivation for applying complex systems thinking to educational systems is that our societies are becoming increasingly complex and intertwined. The modern globalised world needs mainly specialists – people who are particularly good at a few things, which often do not correspond to classical school teachings; some “all-rounders”, who are good at many things, will make connections between them. When educating today's and tomorrow's generations – enabling them to be valuable citizens that contribute to a successful society<sup>1</sup> – the educational system must provide people with a certain minimal common background. Moreover, and potentially even more important, the educational system must help specialists acquire their particular skills and knowledge which will make them the valuable resources of society.

Building and maintaining a well-performing educational system, which is able to cope with varying conditions and stresses given through migration, economical crises, changing professional requirements and other factors, is a very challenging task. Growing difficulties in the educational systems manifest themselves all over the world, and it is time to find innovative solutions.

The author of this paper has thorough knowledge of the Swiss educational system, not only through her own experience, but also because over the last 6 years she has been closely involved with primary schools and in touch with universities of teacher education in Switzerland, through the

<sup>1</sup>Although, according to Davis and Sumara (2006), there is considerable philosophical controversy about the purpose / effect of education. For the scope of this paper, we will assume that the purpose of education is to enable people to become valuable citizens that contribute to a successful society which includes the dignity and welfare of as many citizens as possible.

KIDSinfo project, <http://www.kidsinfo.ch>. It was launched by the Swiss Association of Women Engineers, SVIN, to pique school children's and particularly girls' interest in technology.

*Organisation of this article:* The second section details the typical problems of many educational systems. The third section explains relevant system characteristics. The fourth section looks at scale and complexity found at different levels in the educational system. The fifth section brings up several controversial issues. The sixth section introduces the case study made on the educational system in Switzerland. The seventh section presents related work. Finally, the eighth section draws conclusions and makes further recommendations.

### Problems in educational systems

It is a widely supported observation that in almost all countries – to a varying degree – young people increasingly have problems finding their ways in society. Another indicator of trouble is that in standardised tests applied across one or several countries, such as the PISA tests (the Organisation for Economic Co-operation and Development (OECD) Programme for International Student Assessment), performance is often poor. Well-regarded schools or countries frequently fail to meet the expectations.

Generally speaking, many educational systems are high in cost but low in efficiency, and lots of social problems surface with or without observable trigger. Many schools complain about disruptive behaviour, violence, cheating, students dropping out, etc.

Reforms of the traditional type have shown close to no improvement; multi-billion \$ projects such as “No child left behind” in the USA were abandoned after years of efforts in vain because the large scale actions taken failed to bring the desired effects.

The point is that people have diverse backgrounds, skills and preferences; therefore, a “one fits all” approach does not work well in many cases because it fails to take people's individuality and their different ways of interacting and learning into account. While the “average” students may react well to an “average-fitted” approach, there will always be plenty of students that will not – for instance, because they are overstrained, under-challenged, because their interests are not met, or because they do not understand the importance of education for their life.

In this situation, insights gained from complexity science may help. Such an approach takes into account the importance of scale and complexity at various levels in the educational system, and may help provide the system with suitable tools at the right level.

Learning is itself a highly complex process which involves many different factors and perspectives, such as individual sense-making, teacher-student relationships, class-

room dynamics, school organisations, community involvement, bodies of knowledge, and culture (Davis and Sumara, 2006); knowing, knowing how to do, and knowing how to be (Lelouche and Morin, 1997). Bar-Yam (2004) discusses both the inherent complexity of learning itself as well as the different levels of complexity in educational systems; concrete examples generally refer to education in the USA. Other authors are cited in the related works section of this paper; our main focus here is on the system which provides the students with opportunities for learning while motivating their curiosity and creativity.

### System characteristics

The original Latin word *complexus* signifies *entwined* or *twisted together* (Heylighen, 1996). A complex system is thus made of more than one part, and the parts are at the same time distinct and connected. It is therefore inherently difficult to model them. Often, there are circular causal relationships: one part influences the other, which in turn influences the first, and so on. This description definitely fits educational systems, with their multi-lateral interactions between teachers, students, their parents, families and friends, teachers' and students' associations, politicians, economy, and the society in general.

Due to its distributed nature, the educational system has weak interdependences between individual classrooms and between individual schools. What happens at one local school does not automatically have much to do with what happens at other schools, in other neighbourhoods or other cities. Schools are strongly influenced by local conditions, and within a certain school, what happens inside a certain classroom is strongly dependent on the teacher, the course to be taught, the students, and their parents. This leads to random quality (influenced by many local and some global factors).

Many system behaviours are local and fine scale – at the level of the individual student or teacher and the interactions they engage in with others. The difficulties encountered are often very particular to a certain case; an action successfully taken in one case may fail in another similar case.

Generally speaking, systems with high variety perform well when faced with complex challenges. This means that a system which is itself complex enough and has a variety of ways to address individual problems will be successful when facing a situation of high complexity and variety, as taught by the *Law of Requisite Variety* (Ashby, 1956). This certainly also applies to educational systems and the challenges they must cope with.

Many different ways of learning exist, including visual, auditive, tactile and other stimuli, and “learning by doing”. Learning in diverse ways provides people with diverse ways of addressing challenges, which in turn often triggers innovation and thus economic growth. This means that it should be in the educational system's very best interest to provide

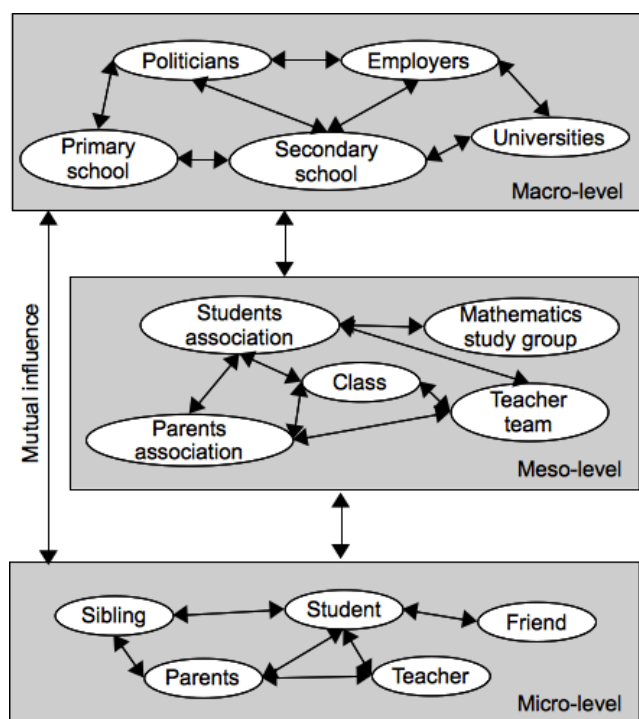


Figure 1: Examples of entities and their interactions at different levels of the educational system (non-exhaustive)

a variety of ways for people to learn. The goal should be to achieve a great variety of skills with consistently high quality, in whatever area of expertise.

The guideline should be to think globally – in terms of the entire educational system and the goals to reach for the benefit of the society – but to act locally – at the level of the individual students, teachers, or groups of them, and taking into account their individual conditions, problems, goals and influencing factors.

### Complexity at different levels

As already mentioned, educational systems typically have several levels of diverse complexity and scale. We discern three of them (although more differences might be made in-between). Figure 1 illustrates this with some examples; other levels and other entities and relations between them exist but have been omitted for the sake of readability.

- **Micro-level / local level:** Education is highly complex at the level of the individual student, his / her capabilities and interests. Many different interrelations are important, including those between student and teacher, student and parents / family / close social environment / other students, and parents and teacher. Actions to take effect at this level must be small-scale and individual. Higher level uniformity of the local tools and actions is not indicated; what works in one case may fail in another.

- **Meso-level / intermediate level:** The complexity at the level of groups of students with similar interests and capabilities is medium, and effective actions can be of medium scale, as they will address student associations, study groups, parents associations or teacher teams. A certain coordination at the meso-level makes sense, as is actions at this level concern groups of people.

- **Macro-level / global level:** Large scale uniform approach can be used at a higher level, including the definition of minimal educational standards for the society to function, teacher education which provides a set of skills for individual action, and teacher support, giving them tools and organisations as required to fulfill their tasks. A lack of coordination or uniformity at the macro-level puts a system in danger of becoming disorganised and confusing.

Taking these differences in scale and complexity into account assures an effective approach to the existing difficulties at each level, because the actions taken are suitable in scale, scope and complexity.

### Issues to be addressed

This section discusses several aspects of importance for a successful education system.

**The right moment for specialisation:** Certain aspects of the educational system are known to be very controversial; among others, the right age for specialisation. It is known from cognitive research that key connections and processes in the brain are established at an early age. This would speak in favour of an early specialisation, so that children would develop their special skills under optimal conditions. On the other hand, children may need enough time to learn a broad variety of skills and develop large general knowledge before even being able to decide which their favourite area shall be. A scientifically sound and generally accepted answer to this topic has not been found yet.

**People's critical attitude towards teachers:** It is an interesting observation that we are highly critical of teachers, but very little critical of medical doctors, although the latter ones have just as much responsibility for our well-being as the former ones, and both of them are human and thus prone to errors. While teachers carry a good part of the responsibility for our positive development on the mid- and longer term, doctors' interventions are often on the shorter term (we typically ask for their help when something is wrong; only few of us consult doctors for advice while everything is fine).

One possible reason for these different perceptions of responsibility and well-being may be that (in most countries) we have a certain freedom to choose who to take as our doctor; if we are not happy with one, we can move on to the

next, until we are satisfied. There is thus a certain competition between doctors of the same specialty. Teachers, however, do not compete with each other, and people have mostly no possibility to choose which one we want to trust with our / our children's education. In some countries, people can choose to which school they want to send their children, but there, the selection possibilities end, and they must accept the teachers they are given. On the other hand, private and higher level schools get to select their students by specifying minimal performance requirements or through other selection procedures.

A way of changing this situation would be to introduce mechanisms for competition and mutual selection of students and teachers in the educational system. The following versions are imaginable:

- Let schools choose students. Let students choose schools. Both versions already exist for private schools and higher / specialised education institutions.
- Let students choose teachers. Let teachers choose students. Neither version does usually exist, to the best of the author's knowledge. If two or more teachers offered exactly the same course, the students' way of choosing a teacher would depend on his / her teaching style and personality. Some teachers might be much more popular than others, and their classes would very quickly be fully booked. They may then either accept students on a "first registered, first served" basis, or select students based on their performances and characteristics. This would naturally lead to inequalities and tension, which may be morally controversial. Competition for scarce resources is, however, also an important principle of how biology and healthy economies function. It would be very interesting to study the effect of such a mechanism on education.

The problem with introducing competition between teachers is that it would add an additional and potentially deceptive performance criterion. Popular teachers are not automatically those which succeed best in transmitting the required knowledge and skills; the most popular teachers may simply be the best entertainers or those which challenge their students less than their fellow teachers. An analogue phenomenon was observed by Ficici and Pollack (1998) when studying the similarities between co-evolution and some educational systems. In their case, each "team" ranks the performances of the other. They showed that this kind of setup can get stuck in mediocre stable states. Indeed, competition may just lead to teachers modifying their content to get good ratings from students. As a consequence, they do well in the competitive environment imposed upon them, but do poorly at fulfilling the wider social goal of producing bright students. This, of course, would be completely contra-productive.

A solution to this problem would be to introduce performance criteria which measure how well teachers reach the main goal – that is, enabling their students to succeed in their further education, and maybe also on the mid- and longer term, that is, in their professional life. Students and their parents would certainly understand that it makes sense to choose a teacher which assures the further academic and/or professional success of their students as opposed to providing good entertainment (although both is relevant for efficient learning).

**Performance evaluations:** An important aspect of the educational system is that it needs to evaluate students. The typically used large scale standard tests fail to thoroughly reflect on people's capabilities and skills. Classical school knowledge cannot be equaled to success in life; very good students may fail in life, whereas weak students may gloriously succeed. Society needs people with a great diversity of individual skills, knowledge and characteristics, including manual skills, emotional intelligence, the ability to collaborate in teams, etc. Often, the required skills and knowledge do not correspond to classical school disciplines, and are thus badly reflected by standard examinations in those disciplines.

An alternative and more sensible way of assessing students may be so-called "portfolio assessment", meaning that not only single written exams count, but also individual and group projects, self-motivated studies, applications scenarios, and other studies. Typical arguments of educational institutions against such forms of assessment is, for instance, that it is more challenging to define success criteria, and that these assessment forms require more human resources, which are often scarce due to financial pressures.

Moreover, not only the standard skills which everybody needs must be assessed, but also and especially the particular skills needed for particular activities. This is called "niche selection" in biology, and it means comparing similar students with each other. The assessment of "non-classical" skills is not straight forward but well worth the effort of elaborating useful metrics.

Yet another aspect of introducing competition and mutual selection to the educational system is that not only students need to be evaluated, but also teachers, respectively the quality of their teaching. This is difficult because there is a lack of real comparison possibilities: it is impossible to test the same students on the same topic, using a different teaching method / a different teacher. Further efforts to find ways of evaluating teachers and teaching are required, in particular related to the issue of "deceptive goals" discussed in the subsection above.

### Case study: Switzerland

In general, Switzerland and the Swiss educational system are doing very well. Although the country has very few

natural resources, the economy is fairly stable and hardly affected by economic crises. The unemployment rate has for many years remained around 3.5 - 4%, which is very low in comparison with other countries. Switzerland faces high amounts of immigration, and its population currently consists of 22% of foreigners, speaking many different languages in addition to the 4 official languages. Nevertheless, the difficulties surging in the educational system are well coped with. Given that many of the actions recommended by the complex systems approach – for instance having high diversity at the right levels – are already implemented in Switzerland and that the systems works well, we can conclude that the approach is valid. A few problems, however, persist in the educational system, and we will look at them subsequently.

Switzerland has a highly diverse educational system which is in many aspects governed by what we call the “Kantönligeist” – the spirit of the little cantons (of which small Switzerland has 26!) – which means that every canton can autonomously decide about their school system. This fragmented attitude is due to the nature of the Swiss state, which is a federation, and due to the populations appreciation of old ways and traditions; conservatism prevails.

Remarkably, the country has almost exclusively public schools, and they generally are of high quality – as just about everything in Switzerland; the few private schools have particular characteristics as following Rudolf Steiner’s teachings or being international / foreign.

One of the strongest points of the Swiss educational system is that many ways lead through education to profession; some of the ways focus on academic achievements, others provide solid manual and profession-specific training. Details are given in the next subsection, followed by an analysis of the Swiss educational system and recommendations derived from the complex systems approach as introduced earlier in this paper.

## Ways to professions in Switzerland

As illustrated in Figure 2, children enter the educational system at the age of 4 or 5, starting with 1-2 years of the recommended but not mandatory public Kindergarten. Some private Kindergarten establishments accept children much earlier, in some cases even as early as from the age of 4 months. The mandatory nine school years start with primary school at age 6 / 7. There are diverse forms of primary school (4-6 years) and intermediary “cycles” (1-3 years under diverse names), after which the adolescents around age 13 either go to some form of secondary school. It takes about 2-3 years and comes in several levels, according to the students’ capabilities. Assessment is continuous, and there are no major final exams at the end of secondary school.

Those students which show sufficiently good performance in primary school and already know that they are headed towards university may attend the so called “Pre-gymnasium”

(2 years), which then leads to the “Gymnasium” (4 years). Several specialties are available, preparing the students for university. Assessment is continuous, and there are also major examinations at the end, called “Matura”. Succeeding them gives direct access to any university in Switzerland – except for medicine, where *numerus clausus* takes place.

For those ending their school education at age 16, there is an excellent way of acquiring well-founded professional qualifications: an “apprenticeship” is a vocational training on the job, accompanied by 1-2 days per week at a specialised professional school. Assessment is continuous and includes both practical and theoretical evaluations. The final examinations lead to a nationally recognised diploma, which is crucial for future employment. About 70% of the adolescents choose this option, which gives them a solid practical education while already receiving a small salary (which is a considerable advantage in comparison with those who are still in full-time school!). In case the young adults with professional diplomas wish to acquire further qualifications, they can either attend technical schools or top-up their education with 1-2 years of general education which leads to a “professional Matura” and gives them access to the universities of applied sciences.

Further information about the educational system in Switzerland is available on:

<http://www.swissworld.org/en/education>.

## Current state of the Swiss educational system

On the **positive** side, as mentioned before, the general state of the educational system in Switzerland is rather good. Already at primary schools, individual support tutoring is offered to students with special needs, no matter what it is: the local language, mathematics, reading, writing, keeping their attention focused, general learning skills, or something else. In some cities, senior citizens accompany school classes several days per week and provide support to teachers and students. Evaluations in most primary schools and some secondary schools include talks between child, parents and teacher for the assessment of the child’s performance and the setting of individual learning targets.

While there are national standards for education, there are also many individual ways of achieving them, based on the student’s characteristics, interests and performance. As it turns out, with this plentitude of possibilities, there really is a way for everybody to receive suitable education.

On the **negative** side, many primary school teachers complain about their massive work load, trouble with parents, too much responsibility, too many diverse objectives to achieve, and difficulties with the coordination of their students very diverse time tables which must include all their support tutoring and special lessons.

Over the last 2 decades, male teachers have become a rarity at primary schools, whereas at secondary schools, the

Approx. age	...diverse ways, ranging from manual professions to academic professions...		
4 - 6	Kindergarden (1-2 years)		
6 / 7-12	Primary school (4-6 years) (+ intermediary "cycles"), diverse names !		
13-15	Secondary school on 3 levels (2-3 years); continuous evaluations but no final examinations		"Pre-gymnasium" (2 years)
16-19	Apprenticeship / vocational training (2-4 years): training on the job + 1-2 days / week at specialised school; continuous practical and theoretical evaluations, final examination gives nationally recognised diploma ...about 70 % of people choose this option! (*)		"Gymnasium" / high school (4 years), diverse specialties; final examinations ("Matura") give university access except for medicine
	Work	Professional "Matura" (1-2 years)	...also other types of schools (nurses, teachers, business)
20 - ...	Work  ...further courses at diverse higher schools	Universities of applied science, B.Sc. (not valid at universities)	Universities

Figure 2: Swiss education system overview

ratio between male and female teachers is still quite equilibrated. It has not been finally determined what the reasons for this development are, but it has been suggested that it may go hand-in-hand with the declining prestige which society attributes to primary school teachers. Again, the reasons are not known, but they may well be related to the previously mentioned total absence of competition and selection possibilities.

Another difficulty which people face in the Swiss educational system are the complications that come with moving from one canton to another, which is in today's dynamic society a rather frequent necessity. The transition from one canton's educational system to another does often not go smoothly. The time when a second language of the country and English are introduced differs considerably, even from one city to another. Some insist that English should be the third language the children learn, and only address it in secondary school, while others start their "early english classes" already in the first year of primary school or even Kindergarden, and before the second language of the country. Similar inconsistencies exist also in the areas of mathematics or natural sciences.

Interestingly, the B.Sc. which the universities of applied sciences award do not provide direct access to M.Sc. studies at academic universities; conversion courses are required. Similarly, people with a classical Matura do not have direct access to the universities of applied sciences; practical experience in industry is necessary. As it is, a B.Sc. from an academic university is thus not equivalent to a B.Sc. from a university of applied sciences. These difficulties are one of the drawbacks of such a diverse system.

## Recommendations of the complex systems approach

Based on the general analysis of the different levels derived from the complex systems approach to educational systems as detailed in the fourth section – entitled "Complexity at different levels" – and the analysis of the Swiss educational system in the above subsections, the following concrete recommendations are made:

- **Micro-level / local level:** The diversity of actions and measures available for supporting the individual student already being very high, the system has sufficient complexity for addressing the diversity of needs. However, the responsibility for the students' education could be distributed over a team of teachers and experts, including psychologists; Davis and Sumara (2006) also suggest this. Forming teams would relieve the currently high pressure on individual teachers, and transfer the coordination of the task from the micro- to the meso-level.
- **Meso-level / intermediate level:** Some elements of competition and selection may both increase the prestige attributed to the teacher profession (and thus its attractiveness for male teachers) and mitigate the critical attitude of the public towards teachers. A way to introduce competition without drastically changing the school system would be to have publicly available teacher and school ratings; maybe a bonus part of the teachers' salaries could depend on their rating by students, parents and peers. Such ratings would need to include both popularity, which is related to entertainment value and freedom of choice, as well as the short-/mid-term achievement of academic

goals and mid-/longer term professional success. More important changes could be made in later steps.

Where necessary, more support for groups of immigrants could be offered. Immigrants not only need to learn the local language but should also familiarise themselves with the local culture to assure a smooth integration.

- **Macro-level / global level:** To improve the consistency of the scholar system, the cantons should finally bring themselves to agree on a common school structure. There is no objectively sensible reason to keep the differences. The HarmoS project<sup>2</sup> aims at this, and about two third of then cantons have accepted to join, but the other third sadly refused. Similarly, it is necessary to agree on when to introduce English and the second language of the country; additional languages are optional and therefore not problematic.

### Related work

The application of the complex systems approach to learning and education has been pursued by a variety of researchers mainly over the last two decades. A working group first met at the NECSI – New England Complex Systems Institute – in 1999. Kaput et al. (1999) state that their intention was to apply the complex systems approach to education in content, teaching, learning, cognition, and the educational system itself. They started by asking the plenty of questions; some answers are given by Bar-Yam (2004).

Lelouche and Morin (1997) emphasise the difference between three education-related knowledge types: knowledge about the domain and problem-solving, which are both to be acquired by the students, and tutoring knowledge, used by the system to facilitate the students' learning process. These three types are modelled at different levels of abstraction, to shed a uniforming light on the educational system's operation and performance.

Vanderstraeten (1997) studies the discrepancy between an economic perspective on the educational system, which focuses on manpower-planning / cost-benefit analysis, and a social perspective, which wishes for an educational system than satisfies the "voice of the people". Both perspectives, however, neglect the fact that education is a composition of complex circular processes between the educational system and society. Policy-makers need to take this into account when designing educational systems.

Davis and Sumara (2010) point out that learning is complex, and education is one of the most complex of human enterprises. Most complex systems are also learning systems. The authors review insights gained by researchers looking into a holistic and action-oriented complexity. Classrooms can be described as knowledge-producing networks, rather than contexts that are centered around a teacher or student.

<sup>2</sup>Information about HarmoS is available in German and French on: <http://www.edk.ch/dyn/11659.php>.

Similarly, curricula should not be seen "in terms of basics and foundations in discrete disciplines, but rather as nodes, hubs and links in decentralised networks of human knowing". Also, learning is not so much the achievement of an individual, but rather something that emerges from the participation and implication of others.

In their book, Davis and Sumara (2006) look into the importance of complexity for various aspects of education, including learning, teaching and research, and suggest complexity thinking as an appropriate attitude for people involved with education. Among other findings, they conclude that teams can considerable out-perform the sum of the team members individual actions. This is a fact which has implications for the classroom, school boards, associations, communities and societies.

Complexity and education has received an increasing amount of interest over the last few years. A rich resource about this topic is <http://www.complexityandeducation.ualberta.ca>. An annual international conference has been held since 2003 under the name of the "Complexity science and educational research conference", and a corresponding journal is published under the name of "Complicity: An International Journal of Complexity and Education".

### Discussion and conclusion

Findings from complexity science can help solve problems in man-made complex systems, including educational systems. A key point is to recognise the importance of complexity and scale at different levels, and to adapt the available instruments, tools and measures to be taken accordingly. For instance, a large scale uniform approach is ill-suited to address a problem which requires diversity at a smaller scale. On the other hand, missing standards at the global level can cause inconsistencies at lower levels and thus lead to unnecessary turbulences.

Some may argue that education is merely complicated and not complex. True, the fact that different people have different learning preferences and abilities is not specifically a complex systems idea, and is probably better described as effectively the theory of individual differences from psychology. Teachers and the educational systems must then compromise to build a system which is reasonably well-suited for most students and provides society with individuals that have the necessary skills and knowledge for society to function. However, taking the working definition (Frei, 2010) that a complex system is composed of many multi-laterally interacting individuals, where changes in one place may have consequences at another place, educational systems very well qualify as being complex. The question investigated in this paper is whether the complex systems approach may provide useful hints at how to improve educational systems, and this has been confirmed so far. Further practical investigations, however, are certainly indicated.



In the case of the Swiss educational system, the great diversity of tools and intervention possibilities at the micro-level allow the teachers and school psychologists to find a suitable approach for every child; responsibility, however, should be distributed among a team instead of being on one teacher alone. At a meso-level, groups and associations for students with similar characteristics and interests would provide support and incentives for maximising performance, both for students and teachers. At the macro-level, Switzerland has a very diverse system in two senses, with only one being helpful: the diversity of ways to a profession, including vocational education in the form of an “apprenticeship”, specialised technical schools, universities of applied sciences, and academic universities, is certainly a strength of the Swiss system and assures the high quality of professionals. On the other hand, the differences in the school systems between the cantons is rather disturbing and hinders people who move from one canton to another from advancing as desired, and more nation-wide uniformity at the macro-level would make sense.

Concrete measures to be taken to improve the educational system in Switzerland include the shifting of responsibility from individual teachers towards small teams which may include psychologists and other experts, and the agreement on a consistent education system structure across the country.

Generally speaking, for a complex adaptive system to function and cope with changing conditions and incidents, a multi-level approach with great diversity at the micro-level, many choices at the meso-level, and common standards at the macro-level is recommended.

Once the importance of complexity and scale at different levels has been recognised for the educational system, the next steps include persuading politicians and authorities in the educational system, which is quite a challenge in itself. The human reluctance to change is considerable, especially since the traditional approach was successful in the past. However, the world is changing rapidly – among others becoming increasingly connected and intertwined – and thus our approach to teaching must change, too.

## Acknowledgements

Regina Frei is currently receiving a fellowship for prospective researchers from the Swiss National Science Foundation.

## References

- Ashby, W. (1956). *An introduction to cybernetics*. Chapman & Hall, London.
- Bar-Yam, Y. (2004). *Making things work: solving complex problems in a complex world*. NECSI Knowledge Press, Cambridge, MA, USA.
- Davis, B. and Sumara, D. (2006). *Complexity and Education - Inquiries into Learning, Teaching and Research*. Lawrence Erlbaum Associates, Mahwah, NJ, USA.
- Davis, B. and Sumara, D. (2010). 'If things were simple...': Complexity in Education. *Evaluation in Clinical Practice*, 16(4):856–860.
- Ficici, S. G. and Pollack, J. B. (1998). Challenges in Coevolutionary Learning: Arms-Race Dynamics, Open-Endedness, and Mediocre Stable States. In Adami, B. and Kitano, T., editors, *6th Int. Conf. on Artificial Life (ALIFE VI)*, pages 238–247, MIT Press, Cambridge MA, USA.
- Frei, R. (2010). *Self-organisation in Evolvable Assembly Systems*. PhD thesis, Department of Electrical Engineering, Faculty of Science and Technology, Universidade Nova de Lisboa, Portugal.
- Heylighen, F. (1996). What is complexity? <http://pespmc1.vub.ac.be/COMPLEXI.html>.
- Kaput, J., Bar-Yam, Y., Jacobson, M., Jakobsson, E., Lemke, J., and Wilensky, U. (1999). Two roles for complex systems in education: mainstream content and means for understanding the education system itself. planning documents for a national initiative on complex systems in k-16 education. Technical report, New England Complex Systems Institute, Boston, MA, USA.
- Lelouche, R. and Morin, J.-F. (1997). Use of abstraction and complexity levels in intelligent educational systems design. In *15th Int. joint conf. on Artificial intelligence - Vol. 1*, pages 329–334, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Vanderstraeten, R. (1997). Circularity, complexity and educational policy planning: a systems approach to the planning of school provision. *Oxford Review of Education*, 23(3):321–332.

# Life as a process of open-ended becoming: Analysis of a minimal model

Tom Froese<sup>1</sup>, Nathaniel Virgo<sup>2</sup> and Takashi Ikegami<sup>1</sup>

<sup>1</sup>Ikegami Laboratory, University of Tokyo, Japan

<sup>2</sup>Centre for Computational Neuroscience and Robotics, University of Sussex, UK

Contact e-mail: [t.froese@gmail.com](mailto:t.froese@gmail.com)

## Abstract

We argue that the phenomenon of life is best understood as a process of open-ended becoming and that this potentiality for continuous change is expressed over a variety of timescales, in particular in the form of metabolism, behavior, development, and evolution. We make use of a minimal synthetic approach that attempts to model this potentiality of life in terms of simpler dissipative structures, using reaction-diffusion systems to produce models that exhibit these characteristics. An analysis of the models shows that its structures exhibit some instances of relevant changes, but we do not consider them open-ended enough to be called alive. Still, the models shed light on current debates about the origins of life, especially by highlighting the potential role of motility in metabolism-first evolution.

## Introduction – The standard view

In the field of synthetic biology there is a widespread optimism that the creation of an entire living cell from scratch is imminent (e.g. Zimmer, 2009; Deamer, 2005; Szostak, et al. 2001). It is hoped that this bio-engineering approach will help to resolve one of the outstanding mysteries of science, namely the origin of life on earth. The mainstream consensus is that the crucial element in the transition from non-living to living matter is the appearance of evolution. Many of the researchers in the field of artificial life, who are studying the origin of life, also share this guiding idea. Their work is thus focused on the question of how best to simulate or chemically engineer the emergence of self-replicating structures (e.g. Rasmussen, et al. 2004; Solé, 2009). Within this general direction of research we can distinguish two relatively distinct traditions in terms of whether they assume the replication of information or the replication of metabolism to be the first factor in evolution.

The *information-first* (a.k.a. ‘replicator-first’)<sup>1</sup> view of life claims that there was *genetic evolution* right at the start of life itself. An extreme version of this view is known as the “RNA world”, which holds that “the first stage of evolution proceeds [...] by RNA molecules performing the catalytic activities necessary to assemble themselves from a nucleotide soup” (Gilbert, 1986, p. 618). However, it is now recognized that this RNA-only view is incomplete, and that the appearance of

Darwinian evolution also requires the compartmentalization of replicating nucleic acids to ensure the segregation of genomes from one another. The field has therefore turned toward the task of incorporating suitable information-carrying molecules into the right kind of vesicle in a way that ensures the reproduction of both (e.g. Hanczyc, et al. 2003), and in a way that allows for competition and differential success (e.g. Chen, et al. 2004). On this updated information-first view, the role of metabolism in the origin of the first living cell is at most a secondary aspect, and perhaps even completely absent. Rather, the essence of life consists of only two components: “fundamentally, a cell consists of a genome, which carries information, and a membrane, which separates the genome from the external environment” (Chen, 2006: 1558).

The *metabolism-first* view of life, on the other hand, claims that the main driving force at the origin of life was *epigenetic evolution*. A radical version of this view holds that the origin of life coincided with the emergence of autocatalytic systems (e.g. Kauffman, 1986), and that under certain conditions some selective pressures could have already been effective at this chemical level (e.g. Fernando and Rowe, 2007; Meléndez-Hevia, et al. 2008). It has also been claimed that “Darwinian competitive exclusion is rooted in the chemical competitive exclusion of metabolism” (Morowitz and Smith, 2007: 58), and that metabolism has played a bigger role than replication in making novelties appear in evolution (Pulseli, et al. 2009).

Similar to the updated information-first view, many of the metabolism-first researchers also argue for the essential role of some kind of spatial separation. It is said that autocatalysis by itself is not sufficient for life, and that these processes must necessarily be part of the constitution of a spatially localized individual (Maturana and Varela, 1980). Some researchers have gone further in claiming that the network of autocatalytic processes must necessarily be enclosed within a bounding membrane (e.g. Luisi and Varela, 1989).

Modeling studies along these lines have tended to assume that a physical membrane is essential, because it prevents the autocatalytic processes from diffusing into the environment (e.g. Bourguin and Stewart, 2004; Varela, et al. 1974), and allows the regulation of molecular intake (e.g. Bitbol and Luisi, 2004). Research in prebiotic chemistry has shown that it is possible to engineer the emergence of membrane-bounded micelles that provide the autocatalysis for their own replication (e.g. Walde, et al. 1994; Bachmann, et al. 1992; see also the model by Ono and Ikegami, 2000). In addition, recent models have demonstrated that under some conditions the growth and division of membrane-bounded autocatalytic

<sup>1</sup> We call the ‘replicator-first’ tradition ‘information-first’ here in to avoid the misleading impression that the ‘metabolism-first’ tradition does not involve replication. The core of the dispute is not about replication versus emergence as such, but rather about what kind of replication was primary, namely informational versus metabolic or compositional.

systems can lead to differential replicative success (e.g. Ono, 2005; Ono, et al. 2008). On this view, which is sometimes identified with the “autopoietic” approach (e.g. Maturana and Varela, 1980; Varela, et al. 1974), the essence of life consists in a membrane-bounded, self-producing system.

It is important to notice that, although the two mainstream traditions may differ in emphasis, they do not hold mutually exclusive theories about the essence of life. In fact, they both accept the general claim that a biological individual is defined by the physical boundary that is imposed by its membrane, although they have different primary reasons for doing so (i.e. unit of selection versus unit of self-production). And they also both accept that life is essentially about stability and survival, and that the driving force of instability and biological change is primarily located outside of the individual, in the external environment and in evolutionary changes. They only disagree on the details of this account (i.e. is survival primarily about other generation or self re-generation, and is the beginning of evolution genetic or epigenetic). In general, the underlying assumption of the mainstream view is that the first form of life is essentially structurally isolated and behaviorally passive.

In this paper we will challenge this assumption. We follow Virgo (2011) in arguing that dissipative structures whose self-production is spatiotemporally localized, but not necessarily membrane-bound, have much in common with living beings. Even very simple examples of these structures are capable of motility, adaptive behavior, structural change, and epigenetic evolution. Consequently we regard such systems as worthy of study in the context of the origins of life.

### **Living without doing? An alternative view**

Despite some outstanding disagreements, the two mainstream traditions are united by a theoretical view of life that is centered on a combination of the spatiotemporal conservation of the individual with an evolutionary realization of biological change. Accordingly, there are promising attempts to bring these two traditions together, such that life is viewed as essentially consisting of three distinct and yet functionally interrelated components: an informational system, a metabolic system, and a compartment (e.g. Rasmussen, et al. 2003; Ganti, 1975). And given this convergence of the two main traditions, and considering the recent experimental successes in realizing this view via synthetic biology, it seems that the optimism pervading the field is well founded. The creation of all kinds of useful artificial life forms appears to be within our grasp, and the final mysteries of the origin and evolution of life on earth seem tantalizingly close to being resolved.

However, the confident promises of synthetic biology will sound all too familiar to those of us who know the history of synthetic psychology – an approach better known as artificial intelligence. Indeed, around half a century ago there was a similar optimism prevalent in the scientific community that the creation of artificial minds and conscious robots was just around the corner. The driving force of that optimism, which in hindsight looks hopelessly naïve and deeply misguided, was a digital-information-centered science of the mind that resonated with advances in engineering and technology.

Today the view that cognitive science can be reduced to computer science is no longer in fashion, although the alternative still remains to be properly worked out (Froese 2010). How ironic it is, then, that at the moment in which

cognitive science is undergoing a major theoretical makeover, namely toward a view of the mind as essentially embodied, embedded, and enactive (e.g. Gallagher 2005; Clark 2008; Thompson 2007), the science of life is at the same time extolling the virtues of trying to reduce the complexities of cellular biology to the abstract linearity of “logic circuits” (Nurse 2008) and “computer programming” (Balazs & Epstein 2009). History, it seems, is repeating itself.

But the purported reduction of life to logic is not as straightforward as the recent advances in biotechnology may seem to indicate. In particular, we note that, in a crucial sense, the life of the individual organism is completely absent from the mainstream framework outlined above. On the one hand we have structural self-maintenance, and on other hand we have informational self-replication. However, we know the former from the general class of dissipative structures, and the latter from the case of viruses – and neither of these two phenomena is typically considered as being alive. What they are missing is the autonomous expression of goal-directed behavior at the level of the individual, namely forms of translational movement and transformational change, which can be studied in terms of ethology and ontogeny.

We propose that all of these aspects of life, i.e. metabolism, behavior, development, and evolution, are integrated into one coherent process of open-ended becoming. On this view, the possibility of distinguishing between these different aspects is simply due to the fact that the process of living is expressed in terms of activities on a variety of timescales. All known forms of life are embedded within four broad categories of change:

Metabolism: the events on this timescale are taking place continuously in the chemical domain. They are foundational in that they realize the concrete, spatiotemporally localized, existence of the individual living being in an autonomous manner via self-production (Barandiaran and Moreno, 2008).

Behavior: the events on this timescale are unfolding in the relational domain of the individual-environment interaction in a moment-to-moment manner. The relational changes can be more or less tightly coupled to metabolic changes (Egbert, et al. 2010), but they are a non-reducible emergent property of the interaction that cannot be conceptualized non-relationally.

Development: events on this timescale make an individual become a structurally qualitatively different kind of individual within its lifetime. Examples are learning and morphogenesis.

Evolution: structurally qualitative changes in the historical lineage of generations of individuals take place on even larger timescales. Examples are genetic, compositional genetic, and epigenetic forms of evolution that are shaped by natural selection, sexual selection, and/or natural drift.

Of course, the differentiation of the changes exhibited by living beings into these four distinct timescales should not be misunderstood in any absolute sense. Our starting point is to treat life as a unified phenomenon, and these distinctions do not reflect strict boundaries between the distinct timescales of becoming. While each of these timescales can be addressed in relative isolation, as demonstrated by their respective fields of scientific study: molecular biology, ethology, developmental biology, and evolutionary biology, a complete understanding of life must be able to show how these different aspects are expressions of one and the same unified phenomenon. They are mutually interdependent and yet non-reducible.

We suggest that one way of approaching this issue is by introducing the intermediate timescales, namely behavior and development, into the current debates surrounding the origin of life. We need to consider that the living ‘self’ referred to by the notions of self-maintenance and self-replication is a center of activity, i.e. an agent (Ruiz-Mirazo, et al. 2010). And at the same time this additional complexity requires a model that is simple enough so that it can still be understood in a complete manner. To be sure, it may be that the most minimal form of life that satisfies our timescale criteria would actually have to be a membrane-bound single-celled organism that is already capable of information-based genetic evolution by means of natural selection. This is, of course, the hope that is harbored by those in synthetic biology who are trying to create life by combining bounded self-maintenance with self-replication.

On the other hand, we know from work in artificial life that some life-like behaviors can already be found in protocells and prebiotic chemistry. For instance, it has been shown that metabolic self-production can easily lead to movement as well as adaptive gradient following, i.e. chemotaxis, in minimal models of protocells (e.g. Suzuki and Ikegami, 2009; Egbert, et al. 2010). Similarly, it has been demonstrated that some of the chemicals typically favored for the synthesis of artificial cells can spontaneously form oil droplets that exhibit self-sustained motility and a type of chemotaxis (e.g. Hanczyc, et al. 2007; Toyota, et al. 2009). It is in this context that there have been calls for the establishment of a new field of study, variously labeled as “homeodynamics” (Ikegami and Suzuki, 2008), “chemo-ethology” (Egbert and Di Paolo, 2009), and “chemical cognition” (Hanczyc and Ikegami, 2010). In what follows we make a novel contribution to this endeavor.

## The primacy of movement

Let us conclude this introduction by outlining our motivation for the rest of this paper. It has been argued that the ‘RNA world’ hypothesis faces considerable difficulties when confronted with the constraints of prebiotic Earth (Shapiro, 2000). One promising response is to reject the requirement of a *digital* genetic system for open-ended evolution, and to relax the distinction between genotype and phenotype. It is possible that these two features may not have been present at the origin of life, but developed in later stages. We therefore assume that a primordial protocell’s chemical mixture itself can serve as a kind of “compositional genome” (Segré, et al. 2000), which remains relatively well preserved during protocell division; or alternatively that heredity can be achieved through multiple attractors in the autocatalytic reaction network’s dynamics, as in the model of Fernando and Rowe (2007).

We could also assume the existence of a self-organizing membrane structure to protect the consistency of the chemical mixture from adverse environmental influences, e.g. a lipid vesicle (Luisi, et al. 1999). This is the main alternative “Lipid world” scenario of the origin of life (Segré, et al. 2001). However, through this additional step the scenario inherits the major underlying assumptions of the standard view, namely that the origin of life gave rise to an essentially structurally *isolated* and behaviorally *passive* entity. The living individual is enclosed in an interactionally inert compartment. And yet all life as we know it today is an active process of organism-environment interaction and its adaptive regulation (Di Paolo, 2009), and the membrane of cellular organisms is an active

interface in this process (Hanczyc and Ikegami, 2010). It is precisely by means of this active self-other interface that a cell regulates its metabolism and behavior through chemical and sensorimotor coupling (Bitbol and Luisi, 2004).

This dilemma leaves us with two possibilities: either we continue to assume that life began enclosed in a compartment and try to explain how this boundary later developed an active role, or we relax the traditional requirement of a compartment as the first step in biological organization (Tanford, 1978). It may seem that only a structural compartment can ensure the individuality of a protocell as an entity that is distinct from its environment, but this is not always the case. This assumption confuses the organizational limits of the organism with its spatial boundaries (Virgo, et al. 2011). It is possible that chemical gradients are sufficient for the self-maintenance of a coherent systemic identity, as we will argue below.

While it is true that such a flexible ‘boundary’ makes it more challenging to survive in unfavorable environmental conditions, it is also the case that some adverse effects of the environment can be mitigated by rapid multiplication and, especially, by motility and directed exploration – a possibility that has not yet been sufficiently considered by the standard view. Here we see the importance of distinguishing between different timescales. In other words, in evolutionary terms it does not matter if these individuals are more prone to die from environmental events, as long as they can replicate and move to different areas quickly enough. The whole population must be sufficiently distributed in space such that some of them always remain alive. It is therefore conceivable that at the origin of life a capacity for adaptive self-motility came before the development of a more solid self-boundary. The model described in the next section is intended as a minimal proof of concept of this possibility.

## Toward a Minimal Model of Life

One of us (Virgo, 2011) has argued that many of the properties of living organisms are shared by simple dissipative structures of the kind that form in reaction-diffusion systems. Prigogine (1955) coined the phrase “dissipative structure” to denote a structure within a physical system that is actively maintained by a flow of energy and/or matter, rather than being an inert structure that is merely resistant to decay. Prigogine observed that living organisms are dissipative structures in this sense; however there are many other examples.

Given what has been argued above, a suitable starting point for our model would be a self-sustaining chemical processes that is a spatiotemporally coherent individual, and yet is non-compartmentalized. These criteria are met by a special class of dissipative structures, which Virgo (2011, Chap. 5) has called *precarious, individuated dissipative structures*. In addition to being dissipative structures, organisms have the properties of being *precarious*, in the sense that if their structure is sufficiently disrupted it will stop being maintained (i.e. death); and *individuated*, in the sense that organisms are spatially localized, and this localization is a result of the dissipative processes that make up the organism, rather than being imposed from outside (see also Di Paolo, 2009).

Virgo points out that certain other dissipative structures share these properties with living organisms. One non-living

example of this type is a hurricane (McGregor and Virgo, 2009). It is dissipative in that it ‘feeds’ off a temperature gradient between the sea surface in the upper atmosphere; it is precarious in that if an important component is removed it can blow out (as will eventually occur if it passes over land); and it is individuated in that it is the cause of its own spatial localization. Not all dissipative structures are precarious or individuated, and not all precarious, individuated dissipative structures share all properties of living systems. Nevertheless, as Virgo argues, studying such structures provides a useful methodology for modeling some of life’s basic properties.

A simple and easy-to-study system that exhibits precarious, individuated dissipative structures is the Gray-Scott reaction-diffusion system, which was first studied in a two-dimensional context by Pearson (1993). This is a simple model of chemical reactions taking place on a surface. The reaction modeled is a simple autocatalytic one,  $A + 2B \rightarrow 3B$ , meaning that when two molecules of  $B$  collide with one of  $A$ , they react to produce a third molecule of  $B$ . A second reaction,  $B \rightarrow P$ , represents the decay of the autocatalyst into an inert product that leaves the system. The molecules  $A$  and  $B$  have a separate concentration at each point on a 2-D surface, represented by  $a$  and  $b$  (the concentration of  $P$  is not modeled). In addition, the ‘food’ molecule  $B$  is fed into every point at a rate proportional to  $1-a$ . This can be thought of as due to the surface being immersed in a solution of  $A$  at a constant concentration of 1.

Finally, in addition to reacting and being added to the system, the two chemical species can diffuse across the surface. Overall this gives rise to the equations

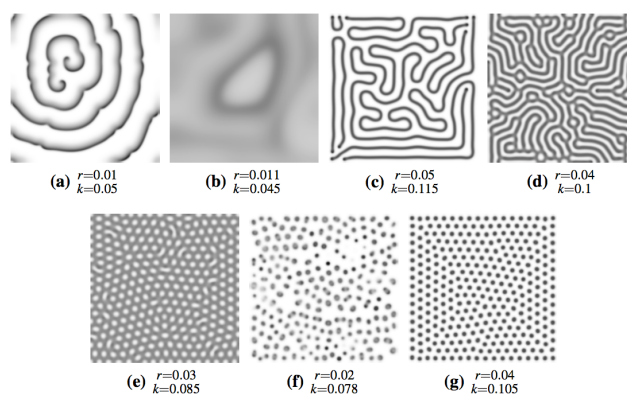
$$\frac{\partial a}{\partial t} = D_A \nabla^2 a - ab^2 + r(1-a); \quad (1)$$

$$\frac{\partial b}{\partial t} = D_B \nabla^2 b + ab^2 - kb, \quad (2)$$

where  $a$  and  $b$  are functions of space as well as time,  $r$  and  $k$  are parameters determined by the rates of the two reactions and the feed process (the rate of the autocatalytic reaction has been set to 1 without loss of generality), and  $D_A$  and  $D_B$  are the rates at which the species diffuse across the surface. These equations can be solved numerically using a method that is akin to a cellular automaton, except that each cell contains a continually variable amount of the two chemical species.

Pearson observed that, depending on the choice of initial parameters, this system can form a variety of patterns, some of which are shown in Figure 1. Of particular interest are the spot patterns in Figure 1(f) and 1(g), since the spots have the properties of being individuated and precarious (Virgo 2011).

Finally, we know that many kinds of dissipative structures that are formed by reaction-diffusion systems are also capable of sustained movement and even replication. This kind of self-organized motility has been investigated experimentally (e.g. Lee and Swinney, 1995; Lee, et al. 1993; 1994) and modeled mathematically (e.g. Varea, et al. 2007; Krischer and Mikhailov, 1994; Pearson 1993). The dynamics of replicating reaction-diffusion patterns have also been studied (e.g. Reynolds, et al. 1994; 1997). In the dissipative structures of the Gray-Scott model we find cases of motility and replication as well, and this includes some kinds of spots. We thus have all the basic requirements to begin our investigation of these spots as a potential minimal model of life as a form of open-



ended becoming, as it is expressed on the four timescales of metabolism, behavior, development, and evolution.

**Figure 1.** Examples showing the range of patterns exhibited by the Gray-Scott reaction-diffusion system with various parameters ( $D_A = 2 \times 10^{-5}$  and  $D_B = 10^{-5}$  in each). The integration method and initial conditions are similar to those used by Pearson (1993). Patterns are chosen as exemplars of various phenomena; see Pearson (1993) for a more systematic classification. (a) A spiral pattern; (b) A chaotic pattern of travelling waves; (c) A line pattern. Lines grow at the ends and then bend to fill space in a process reminiscent of a river meandering; (d) A labyrinth pattern; (e) A hole pattern; (f) A pattern of unstable spots, whose population is maintained by a balance between reproduction and natural disintegration; (g) A stable spot pattern. Spots reproduce to fill the space and then slowly migrate into the more-or-less organized pattern shown (with a different choice of parameters, spots can be produced that are stable but cannot reproduce).

## Metabolism

A reaction-diffusion spot can spontaneously emerge under appropriate conditions, and once it exists, it can self-maintain its precarious existence by means of a continuous turnover of chemical reactions. As a self-producing network of chemical processes it satisfies the requirements of the first timescale. It also provides the reference point of a spatiotemporal entity against which changes on other timescales can be measured.

It is interesting to note in this regard that the spatiotemporal boundaries of a spot are intrinsically fuzzy. It is just as impossible to pinpoint the precise moment in time when the spot begins or ceases to exist, as the precise point in space where the spot ends and the environment begins. This is because the spot is a self-organizing phenomenon that is both continuous in time (temporal ambiguity) and continuous in space (spatial ambiguity). Nevertheless, an intuitive grasp of what constitutes an individual spot is possible; we either see an individual spot on the surface or we do not.

Once an individual spot has spontaneously formed, it will continue to exist even when it encounters a limited range of conditions that would not have enabled its original emergence. The fact that spots can exist outside of their original range of emergence is an indication that they are actively re-producing the viability conditions required for their existence, which can be considered as a strong criterion for autopoietic autonomy



(Froese and Stewart, 2010). It is no different in the case of living beings: although they must have first emerged when the environmental conditions were right, they must now actively produce their own conditions of existence in order to persist.

## Behavior

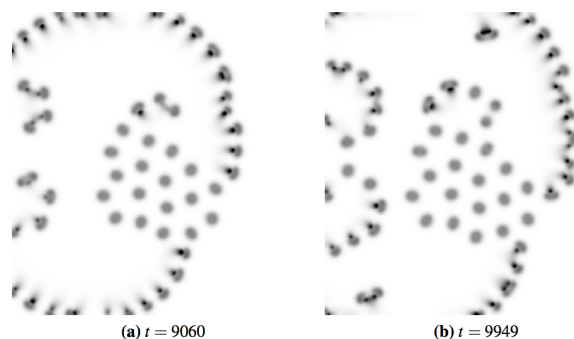
We define the concept of behavior broadly as any change in the individual-environment relationship, which is induced by an instability or tension in that relationship. A behavior ceases when that tension is resolved or transformed into a different kind of tension, which elicits a different kind of behavior. In this paper we take the view that all behavior is characterized by an essential asymmetry centered on the individual (Barandiaran, et al. 2009). The tension that triggers a behavior may originate in the environment, but the fact that there is a response at all is an achievement of the self-constitution of the individual. In this sense their behavior is intrinsically active.

The term ‘behavior’ covers a huge variety of changes in all kinds of entity-environment relations, so some distinctions are in order. One important distinction in biology and psychology is between *reactive behavior*, namely behavior that is triggered by events in the environment, and *active*, or *intrinsic behavior*, namely behavior that is initiated by the individual. Again, the distinction is not an absolute one since, on the one hand, all biological systems have internal state and their reactive behavior is therefore always also a function of their history, and, on the other hand, the expression of active behavior always takes place in the context of environmental events. Nevertheless, a behavior can be more or less driven by autonomous and environmental conditions. Let us consider these two kinds of behaviors in the case of the spots.

**Reactive behavior.** The spots exhibit a clear type of reactive behavior with respect to differences in chemical gradients in their surroundings. We can describe this behavior in terms of approach and avoidance: the spots are capable of following chemical gradients that increase the concentration of their constituents, i.e. chemotaxis, and they are also capable of avoiding chemical gradients that decrease the concentration of their constituents. For example, when we remove constituents from nearby a spot by using a virtual pipette, the spot will tend to move away from the pipette. In this way it is possible to chase spots around the surface. If the pipette is too fast and gets too close to a spot, it destabilizes the spot in such a way that it is no longer sustainable and dies.

If there are several spots in the environment, then these approach and avoidance behaviors will make them interact in certain ways. This is because a spot consumes the food in its proximity, thereby surrounding itself with a negative gradient that keeps other spots away. If the spots did not tend to move away from one another then they would merge rather than remaining separate; these approach and avoidance behaviors therefore form an important part of the individuation process.

Note that although these behaviors are reactive in the sense that they do not occur except in the presence of an appropriate environmental trigger, they are the result of an active growth process. The spot moves because the autocatalyst grows faster on the side where the food concentration is higher. This behavior could thus be said to be reactive in the behavioral domain, but active in the metabolic domain. In order for the spot to move even in the absence of environmental triggers it



**Figure 2.** Two snapshots of the system resulting from Equations (4)-(6), integrated on a surface of 2 by 2 units, with the parameters  $D_A = 2 \times 10^{-5}$ ,  $D_B = 10^{-5}$ ,  $D_C = 10^{-6}$ ,  $r = 0.0347$ ,  $k_1 = 0.2$ ,  $k_2 = 0.8$  and  $k_3 = 0.005$ . The colors are adjusted so that the secondary autocatalyst  $C$  appears as a darker shade of gray than the primary autocatalyst  $B$ . A group of spots with tails can be seen on the mid-left side of plot (a), and after duplication in plot (b) in the same place. Some tail-less spots can be seen as well, their tails having been lost in the process (hence, this is limited heredity with variation). The spots with tails move constantly in the direction facing away from their tails at a rate of approximately  $4 \times 10^{-4}$  distance units per time unit, which results in their colonizing the empty part of space more rapidly than the tail-less spots. However, with this choice of parameters, the tailed spots cannot invade areas occupied by tail-less spots, and they are eventually crowded out and become extinct.

must create its own instabilities. Of course, the whole spot is already in a far-from-equilibrium state, but what is needed is an asymmetrical distribution in the general field of individual-environment relationships (Matsuno, et al. 2007).

**Intrinsic behavior.** One way of achieving active motion is by modifying the original Gray-Scott reaction-diffusion system by introducing a second autocatalyst to the system, which feeds not on the ‘food’ molecule but on the other autocatalyst (see Virgo, 2011). That is, the reactions  $B + 2C \rightarrow 3C$  and  $C \rightarrow P$  are added to the system, so that Equations 1 and 2 are extended to Equations 4-6, where  $D_C$  is the rate of diffusion of  $C$ , and  $k_1$ ,  $k_2$  and  $k_3$  are the rate constants for the reactions  $B \rightarrow P$ ,  $B + 2C \rightarrow 3C$  and  $C \rightarrow P$ , respectively.

$$\frac{\partial a}{\partial t} = D_A \nabla^2 a - ab^2 + r(1 - a); \quad (4)$$

$$\frac{\partial b}{\partial t} = D_B \nabla^2 b + ab^2 - k_1 b - k_2 bc^2; \quad (5)$$

$$\frac{\partial c}{\partial t} = D_C \nabla^2 c + k_2 bc^2 - k_3 c, \quad (6)$$

With an appropriate choice of parameters, the effect of this is to produce spots of the primary autocatalyst, which are accompanied by a region of the secondary autocatalyst. Since the secondary autocatalyst feeds on the primary one, the spot of primary autocatalyst tends to avoid it by moving away, while the secondary spot follows. This gives the secondary autocatalyst the appearance of being attached as a ‘tail’ behind the primary spot (see Figure 2.) The spot-tail system as a whole moves around spontaneously even in a homogeneous

environment. In the sense that this motility depends on the internal constitution of the whole spot-tail system itself, we can characterize it as intrinsic rather than as reactive.

Although this spot-tail system is not strictly speaking an autocatalytic “hypercycle” (Eigen 1971), because the catalytic dependency is not mutual, it nevertheless can be considered as symbiotic to some extent (see Lee, et al. 1997). While the tail is somewhat parasitic on the primary spot (since it contributes nothing to it metabolically), their jointly induced movements can be adaptive in some environments. Thus, in contrast to the standard view that parasitic reactions are a significant problem for the metabolism-first approach because of their detrimental metabolic effects (and hence, the necessity of a compartment, see Takeuchi and Hogeweg, 2009), we argue that this is not always the case. With certain parameter settings, the spot-tail systems can reproduce more rapidly than spots without tails, and their movement also tends to make them colonize new areas more rapidly. This highlights once more the importance of distinguishing between different timescales: what may be detrimental on the metabolic timescale (parasitic reaction), can induce changes on the behavioral timescale (exploratory behavior), which are adaptive on the evolutionary timescale. Figure 3 shows an example of a scenario where over longer timescales spots with tails are better adapted than tail-less spots. The parasite-enabled exploratory behavior helps to prevent the occasional localized extinction events from killing the population. We will return to this finding later.

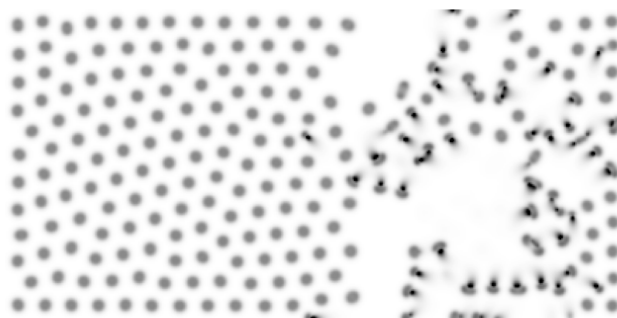
## Development

We conceive of the notion of development in a broad way so as to include any structural changes induced by the organism, which turn it into a qualitatively different kind of being in its own lifetime. These structural changes can include (in order of increasing temporal scale) growth, habituation, learning, adaptation, and ontogeny. Not all forms of life will exhibit all of these variations of becoming to the same extent, but all will display some capacity for developmental change.

We find lifetime dependent structural changes in the case of the spots as well. These changes typically proceed via the incorporation of external elements rather than the internal differentiation that is familiar from modern cells, but we can perhaps still think of this as a kind of proto-development. The emergence of spot-tail systems that was described above is one example. Virgo (2011) also observed a second, related kind of process in a reaction-diffusion system (with a different set of equations), whereby two nearby spots consisting of mutually complementary catalytic reactions join together to form a multi-spot system, thus forming a proper hypercycle (Eigen, 1971). In some respects, development can be seen in single spots as well. When they exhibit directional movement, they do it because they grow toward the increasing gradient, and die back on the other side. They are like plants in that growth and behavior are not always readily separable.

## Evolution

We have already observed that there is a heritable difference between a spot with a tail and a spot without tail (see Figure 2). However they are clearly lacking a digital genetic system with which to encode these differences. In our analysis of the evolutionary capacity of the reaction-diffusion systems we



**Figure 3.** A snapshot from the same system shown in Figure 2, with the same parameters, except that randomly chosen areas in the right-hand side of the surface are occasionally cleared by an externally induced cataclysm (e.g. the food concentration in a random 0.5-by-0.5 area is set to zero every 1000 time units). The spots with tails are able to persist in this region due to their ability to colonize the cleared areas more rapidly than the spots without tails. But in the left-hand side of the figure they are out-competed.

therefore focus only on the possibilities of epigenetic evolution and of evolution with a compositional genome.

**Epigenetic evolution.** It is well known that one of the main epigenetic factors of inheritance is the particular time-space configuration in which an individual is born. A famous case is the beaver's dam, which, once constructed, provides a home for subsequent generations. This kind of inheritance can also occur in the case of reaction-diffusion spots. For instance, the offspring of those spots, which happened to divide because of a high concentration of nutrients, will also find themselves in a situation with high concentration of nutrients.

**Composition-genomic evolution.** We have noted above that the chemical composition of spot can be considered as both its phenotype and genotype combined. The idea is that this kind of ‘compositional genome’ could have enabled protocellular evolution by means of natural selection even in the absence of a digital information-carrying component such as RNA and DNA (Segre and Lancet 2000). For instance, Virgo (2011) has observed spots undergoing a Lamarckian form of evolution, whereby traits that have been acquired during an individual's lifetime are passed along to the offspring. This is the case for spots with tails. Once a spot has acquired a tail (perhaps by passing near to another tailed spot), it will divide in a way that typically results in offspring that have tails.

We also find a difference in selective pressure since in some environments the spots with tails are more viable than the single spots on their own (see Figure 3). This is because their combination results in an internal instability that makes the spot system move around even in the absence of chemical gradients, and they are thereby able to minimize the impact of catastrophic events. Greater spatial distribution lessens overall risk to the population. In this scenario the original single-spot constituents may therefore die out eventually, while the spot-tail variant persists. Here we therefore have all the elements of evolution as it is standardly conceived, namely reproduction, variance, and selection, but with limited rather than unlimited heredity (*sensu* Szathmary and Maynard Smith, 1997).



## Discussion

The model has served as a proof of concept that even simple reaction-diffusion spots can exhibit many essential life-like characteristics, where life is conceived as a process of open-ended becoming. We have focused on the importance of self-organized motility and behavior in the context of current debates on the origin of life. In this discussion we would like to draw attention to the shortcomings of the current model, and to consider possible ways of overcoming them.

The spots satisfied the basic requirements of metabolism (self-creation) and movement (self-motility). In fact, they are even capable of adaptive behavior that resembles the foraging behavior of actual bacteria (nutrient gradient following). The spots are also capable of some proto-development through the incorporation of new external elements, and these lifetime changes are inheritable over generations. Taken together these findings suggest that the spots meet the criteria of undergoing changes within the four major timescales characteristic of life, namely metabolism, behavior, development, and evolution.

But are these spots a model of the phenomenon of life? We characterized life as an open-ended process of becoming, and it is precisely in relation to open-endedness that the limitations of the model are most apparent. How far can this approach be scaled up? Are compositional genomes capable of “unlimited heredity” (Szathmary and Maynard Smith, 1997) as suggested by the work of Segre and Lancet? Is it possible to set up the environmental conditions such that a more complex network of dissipative structures emerges? By which mechanism could such a network learn? How could it reproduce itself?

One issue that would need to be tackled in future models of this kind is how to introduce the possibility of solidity. In the current model the spots are fully transparent to environmental interactions, although chemical gradients may constitute some boundaries. This extreme openness effectively turns the whole spot into an interface with its environment. In order to enable a more open-ended increase of complexity it may eventually become necessary for the system to localize these interfaces at its spatial boundaries. Some researchers have argued that internal differentiation between the constitutive elements that are responsible for self-creation and those that are needed for interaction is a first step toward more behavioral autonomy (Barandiaran and Moreno, 2008). Internal differentiation may enable further specialization of these elements, since they no longer need to do both tasks at the same time.

Relatedly, it is possible that at some point a differentiation between phenotype and genotype may become necessary in order for further evolutionary transformations to become a stable possibility. And even during the organism's lifetime the internal mediation between phenotype and genotype entails a certain lack of self-coincidence in the being of the organism that could facilitate open-ended becoming. The organism's being is then no longer simply a product of its own doing, as it is in the case of the spots, but also of its own genetic self-interpretation. This is because the same DNA can give rise to different expressions in the context of a different phenotype. It is of general interest to further determine to what extent DNA is necessary for the phenomenon of life. One way to address this issue, and which we have pursued in this paper, is to see how far it is possible to get without DNA or any other genetic system. By following this approach some constraints may

become apparent for which a dedicated digital genetic system is an essential part of the solution.

## Conclusion

We have argued that the phenomenon of life is a process of open-ended becoming, and that contemporary debates about the origins of life should take the role of self-organized motility and behavior into account. We revisited Virgo's (2011) arguments concerning simple dissipative structures in reaction-diffusion systems from this theoretical perspective, and discussed the potential of some of these structures as a minimal model of life. We conclude that the current model is able to partially satisfy the proposed view by exhibiting some changes on the temporal scales of metabolism, behavior, development, and evolution. The model also demonstrated the importance of distinguishing between the organizational limits of the organism and its spatial boundaries, as well as between its various timescales. Future work should try to determine to what extent this approach is able to scale up to more complex phenomena, including individuals that have the potential for a greater variety of becoming.

**Acknowledgments.** Tom Froese's research is funded by the Japanese Society for the Promotion of Science (JSPS) and is financially supported by their Grant-in-Aid program.

## References

- Bachmann, P. A., Luisi, P. L. and Lang, J. (1992). Autocatalytic self-replicating micelles as models for prebiotic structures. *Nature*, 357: 57-59.
- Balazs, A. C. and Epstein, I. R. (2009). Emergent or Just Complex? *Science*, 325: 1632-1634.
- Barandiaran, X., Di Paolo, E. A. and Rohde, M. (2009). Defining Agency: Individuality, Normativity, Asymmetry, and Spatio-temporality in Action. *Adaptive Behavior*, 17(5): 367-386.
- Barandiaran, X. and Moreno, A. (2008). Adaptivity: From Metabolism to Behavior. *Adaptive Behavior*, 16(5): 325-344.
- Bitbol, M. and Luisi, P.L. (2004). Autopoiesis with or without cognition: defining life at its edge. *Journal of the Royal Society Interface*, 1(1): 99-107.
- Bourgine, P. and Stewart, J. (2004). Autopoiesis and Cognition. *Artificial Life*, 10(3): 327-345.
- Chen, I. A. (2006). The Emergence of Cells During the Origin of Life. *Science*, 314: 1558-1559.
- Chen, I. A., Roberts, R. W. and Szostak, J. W. (2004). The Emergence of Competition Between Model Protocells. *Science*, 305: 1474-1476.
- Clark, A. (2008). *Supersizing the Mind: Embodiment, Action, and Cognitive Extension*. Oxford University Press, New York, NY.
- Deamer, D. (2005). A giant step towards artificial life? *Trends in Biotechnology*, 23(7): 336-338.
- Di Paolo, E. A. (2009). Extended life. *Topoi*, 28(1): 9-21.
- Egbert, M. D., Barandiaran, X. E. and Di Paolo, E. A. (2010). A Minimal Model of Metabolism-Based Chemotaxis. *PLoS Computational Biology*, 6(12): 1-17.
- Egbert, M. D. and Di Paolo, E. (2009). Integrating Autopoiesis and Behavior: An Exploration in Computational Chemo-Ethology. *Adaptive Behavior*, 17(5): 387-401.
- Eigen, M. (1971). Selforganization of matter and the evolution of biological macromolecules. *Naturwissenschaften*, 58: 465-523.
- Fernando, C. and Rowe, J. (2007). Natural selection in chemical evolution. *Journal of Theoretical Biology*, 247: 152-167.

- Froese, T. (2010). From Cybernetics to Second-Order Cybernetics: A Comparative Analysis of Their Central Ideas. *Constructivist Foundations*, 5(2): 75-85.
- Froese, T. and Stewart, J. (2010). Life after Ashby: Ultrastability and the autopoietic foundations of biological individuality. *Cybernetics & Human Knowing*, 17(4): 83-106.
- Gallagher, S. (2005). *How the Body Shapes the Mind*. Oxford University Press, New York, NY.
- Ganti, T. (1975). Organization of chemical reactions into dividing and metabolizing units: the chemotons. *BioSystems*, 7: 15-21.
- Gilbert, W. (1986). The RNA world. *Nature*, 319: 618.
- Gray, P. and Scott, S. K. (1983). Autocatalytic reactions in the isothermal, continuous stirred tank reactor: isolas and other forms of multistability. *Chemical Engineering Science*, 38(1): 29-43.
- Hanczyc, M. M., Fujikawa, S. M. and Szostak, J. W. (2003). Experimental Models of Primitive Cellular Compartments: Encapsulation, Growth, and Division. *Science*, 302: 618-622.
- Hanczyc, M. M. and Ikegami, T. (2010). Chemical Basis for Minimal Cognition. *Artificial Life*, 16: 233-243.
- Hanczyc, M. M., Toyota, T., Ikegami, T., Packard, N. and Sugawara, T. (2007). Fatty Acid Chemistry at the Oil-Water Interface: Self-Propelled Oil Droplets. *Journal of the American Chemical Society*, 129: 9386-9391.
- Ikegami, T. and Suzuki, K. (2008). From homeostatic to homeodynamic Self. *BioSystems*, 91(2): 388-400.
- Kauffman, S. (1986). Autocatalytic Sets of Proteins. *Journal of Theoretical Biology*, 119: 1-24.
- Krischer, K. and Mikhailov, A. (1994). Bifurcation to Traveling Spots in Reaction-Diffusion Systems. *Physical Review Letters*, 73(23): 3165-3168.
- Lee, D. H., Severin, K., Yokobayashi, Y. and Ghadiri, M. R. (1997). Emergence of symbiosis in peptide self-replication through a hypercyclic network. *Nature*, 390: 591-594.
- Lee, K. J., McCormick, W. D., Ouyang, Q. and Swinney, H. L. (1993). Pattern Formation by Interacting Chemical Fronts. *Science*, 261: 192-194.
- Lee, K.-J., McCormick, W. D., Pearson, J. E. and Swinney, H. L. (1994). Experimental observation of self-replicating spots in a reaction-diffusion system. *Nature*, 369: 215-218.
- Lee, K. J. and Swinney, H. L. (1995). Lamellar structures and self-replicating spots in a reaction-diffusion system. *Physical Review E*, 51(3): 1899-1915.
- Luisi, P. L. and Varela, F. J. (1989). Self-Replicating Micelles - A Chemical Version of a Minimal Autopoietic System. *Origins of Life and Evolution of the Biosphere*, 19: 633-643.
- Luisi, P. L., Walde, P. and Oberholzer, T. (1999). Lipid vesicles as possible intermediates in the origin of life. *Current Opinion in Colloid & Interface Science*, 4: 33-39.
- Matsuno, H., Hanczyc, M. M. and Ikegami, T. (2007). Self-maintained Movements of Droplets with Convection Flow. In Randall, M., Abbass, H. A. and Wiles, J., editors, *ACAL 2007*, pages 179-188. Springer Verlag, Berlin, Germany.
- Maturana, H. R. and Varela, F. J. (1980). *Autopoiesis and Cognition: The Realization of the Living*. Kluwer.
- McGregor, S. and Virgo, N. (2011). Life and Its Close Relatives. In Kampis, G., Karsai, I. and Szathmari, E., editors, *ECAL 2009*, pp. 230-237, Springer Verlag, Berlin.
- Meléndez-Hevia, E., Montero-Gómez, N. and Montero, F. (2008). From prebiotic chemistry to cellular metabolism - The chemical evolution of metabolism before Darwinian natural selection. *Journal of Theoretical Biology*, 252: 505-519.
- Morowitz, H. and Smith, E. (2007). Energy Flow and the Organization of Life. *Complexity*, 13(1): 51-59.
- Nurse, P. (2008). Life, logic and information. *Nature*, 454: 424-426.
- Ono, N. (2005). Computational studies on conditions of the emergence of autopoietic protocells. *BioSystems*, 81: 223-233.
- Ono, N. and Ikegami, T. (2000). Self-maintenance and Self-reproduction in an Abstract Cell Model. *Journal of Theoretical Biology*, 206: 243-253.
- Ono, N., Madina, D. and Ikegami, T. (2008). Origin of Life and Lattice Artificial Chemistry. In Rasmussen, S. et al., editors, *Protocells: Bridging Nonliving and Living Matter*, pp. 197-212, MIT Press.
- Pearson, J. E. (1993). Complex Patterns in a Simple System. *Science*, 261(5118): 189-192.
- Prigogine, I. (1995). *Introduction to the Thermodynamics of Irreversible Processes*. Wiley Interscience.
- Pulselli, R. M., Simoncini, E., and Tiezzi, E. (2009). Self-organization in dissipative structures: A thermodynamic theory for the emergence of prebiotic cells and their epigenetic evolution. *BioSystems*, 96: 237-241.
- Rasmussen, S., Chen, L., Deamer, D., Krakauer, D. C., Packard, N. H., Stadler, P. F. and Bedau, M. A. (2004). Transitions from Nonliving to Living Matter. *Science*, 303: 963-965.
- Rasmussen, S., Chen, L., Nilsson, M. and Abe, S. (2003). Bridging Nonliving and Living Matter. *Artificial Life*, 9: 269-316.
- Reynolds, W. N., Pearson, J. E. and Ponce-Dawson, S. (1994). Dynamics of Self-Replicating Patterns in Reaction Diffusion Systems. *Physical Review Letters*, 72(17): 2797-2800.
- Reynolds, W. N., Ponce-Dawson, S. and Pearson, J. E. (1997). Self-replicating spots in reaction-diffusion systems. *Physical Review E*, 56(1): 185-198.
- Ruiz-Mirazo, K., Peretó, J., and Moreno, A. (2010). Defining Life or Bringing Biology to Life. *Origins of Life and Evolution of Biospheres*, 40(2), 203-213.
- Segré, D., Ben-Eli, D. and Lancet, D. (2000a). Compositional genomes: prebiotic information transfer in mutually catalytic non-covalent assemblies. *Proc. Natl. Acad. Sci. USA*, 97: 4112-4117.
- Segré, D., Ben-Eli, D., Deamer, D. and Lancet, D. (2001). The lipid world. *Origins of Life and Evolution of the Biosphere*, 31: 119-145.
- Shapiro, R. (2000). A replicator was not involved in the origin of life. *IUBMB Life*, 49: 173-176.
- Solé, R. V. (2009). Evolution and self-assembly of protocells. *The International Journal of Biochemistry & Cell Biology*, 41: 274-284.
- Suzuki, K. and Ikegami, T. (2009). Shapes and Self-Movement in Protocell Systems. *Artificial Life*, 15: 59-70.
- Szathmari, E. and Maynard Smith, J. (1997). From replicators to reproducers: the first major transitions leading to life. *Journal of Theoretical Biology*, 187: 555-571.
- Szostak, J. W., Bartel, D. P. and Luisi, L. (2001). Synthesizing life. *Nature*, 409: 387-390.
- Takeuchi, N. and Hogeweg, P. (2009). Multilevel Selection in Models of Prebiotic Evolution II: A Direct Comparison of Compartmentalization and Spatial Self-Organization. *PLoS Computational Biology*, 5(10): 1-17.
- Tanford, C. (1978). The hydrophobic effect and the organization of living matter. *Science*, 200: 1012-1018.
- Thompson, E. (2007). *Mind in Life: Biology, Phenomenology, and the Sciences of Mind*. Harvard University Press, Cambridge, MA.
- Toyota, T., Maru, N., Hanczyc, M. M., Ikegami, T. and Sugawara, T. (2009). Self-Propelled Oil Droplets Consuming "Fuel" Surfactant. *Journal of the American Chemical Society*, 131: 5012-5013.
- Varea, C., Hernández, D. and Barrio, R. A. (2007). Soliton behaviour in a bistable reaction diffusion model. *Journal of Mathematical Biology*, 54: 797-813.
- Varela, F. J., Maturana, H. R. and Uribe, R. (1974). Autopoiesis: The organization of living systems, its characterization and a model. *BioSystems*, 5: 187-196.
- Virgo, N. (2011). *Thermodynamics and the Structure of Living Systems*. DPhil thesis. University of Sussex: UK.
- Virgo, N., Egbert, M. and Froese, T. (2011). The Role of the Spatial Boundary in Autopoiesis. In Kampis, G., Karsai, I. and Szathmari, E., editors, *ECAL 2009*, Springer Verlag, pp. 234-241, Berlin.
- Walde, P., Wick, R., Fresta, M., Mangone, A. and Luisi, P. L. (1994). Autopoietic Self-Reproduction of Fatty Acid Vesicles. *Journal of the American Chemical Society*, 116: 11649-11654.
- Zimmer, C. (2009). On the Origin of Life on Earth. *Science*, 323: 198-199.

# Urdar - an artificial ecology platform

Philip Gerlee<sup>1</sup> and Torbjörn Lundh<sup>2</sup>

<sup>1</sup> Cancer Center Sahlgrenska, University of Gothenburg, Sweden

<sup>2</sup> Mathematical Sciences, Chalmers and University of Gothenburg, Sweden  
[torbjorn.lundh@chalmers.se](mailto:torbjorn.lundh@chalmers.se)

## Abstract

Cross-feeding interactions are a common feature of many microbial systems, such as colonies of *E. coli* grown on a single limiting resource. We have studied this phenomenon in Gerlee and Lundh (2010) from an abstract point of view by considering artificial organisms which metabolise binary strings from a shared environment. The organisms are represented as simple cellular automaton rules and the analog of energy in the system is an approximation of the Shannon entropy of the binary strings. Only organisms which increase the entropy of the transformed strings are allowed to replicate. This system exhibits a large degree of species diversity, which increases when the flow of binary strings into the system is reduced.

## Introduction

The origin of biodiversity has been a long standing problem in ecology and the evolution and maintenance of diversity was long difficult to account for, especially in the light of the proposed competitive exclusion principle which states that several species competing for the same resources cannot co-exist. Related to these issues is the question of how species diversity influences ecosystem productivity (Waide et al., 1999). Several experiments and theoretical models have been devised to resolve this issue, but many of the results have been inconclusive and even contradictory.

One of the simplest ecological system where diversity emerges, and is stably maintained, is in populations of *E. coli* growing in a homogeneous environment limited by a single resource, usually glucose. The diversity is facilitated by cross-feeding (syntrophy), where one strain partially degrades the limiting resource into a secondary metabolite which is then utilised by a second strain. This phenomenon was first observed by Helling et al. (1987).

In Gerlee and Lundh (2010), we present a more general model of the evolution of cross-feeding, which is not aimed at modelling a specific biological system, but rather extracts and models the general principles governing systems where cross-feeding might emerge. In order to do this, we have devised a novel Artificial Life system, named *Urdar*<sup>1</sup> in which

<sup>1</sup>Urdarbrunnr is one of the three wells that lie beneath the world

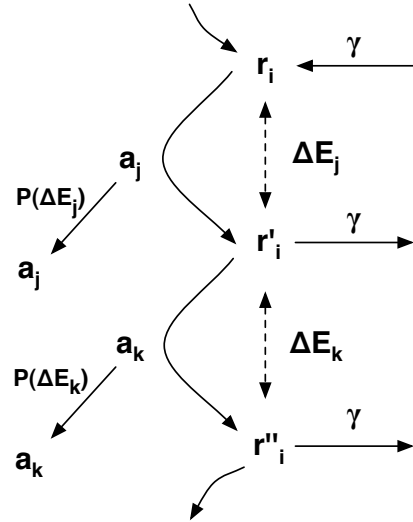


Figure 1: A schematic view of the model. The agents  $a$  in the model digest binary strings  $r$  by applying CA-rules, transforming  $r$  to  $r'$ . To each such metabolic step we can associate a difference in energy  $\Delta E$  (visualised with dotted lines). The reproduction of each agent depends on how much it can decrease the energy of the binary string and occurs with probability  $P(\Delta E)$  (represented by the arrows on the left hand side). The binary strings exist in a common pool which they enter (and leave) at a rate  $\gamma$ , as shown by the arrows on the right hand side.

the fitness of an organisms is defined in a more general sense and where interactions between organisms are at the core of the model. The fitness of the organisms in this model is directly related to their ability to extract energy from a common environment, and is thus closely connected to the fundamental concept of energy which drives many ecological interactions.

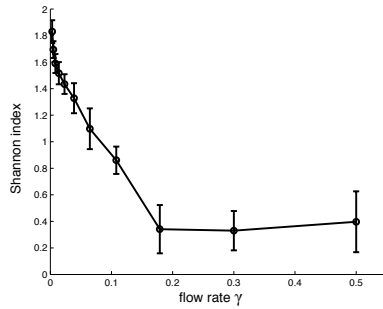


Figure 2: The Shannon diversity index of the species distribution as a function of the flow rate  $\gamma$ . Each data point was averaged over 20 simulations and the error bars represent one standard deviation.

### The Model

In order to give a short description<sup>2</sup> let us give its main features. The dynamics, depicted schematically in figure 1, during one update can be described in the following way:

1. Each agent in the population picks randomly a resource string  $r_j$  from the resource pool  $R$  and transform it accordingly to its CA-rule and then puts the transformed string back into the resource pool.
2. The efficiency of the “metabolic process” just occurred is evaluated by measuring the energy difference  $\Delta E$  of the string before and after the “digestion/transformation”. This is done by drawing a random number  $x$  uniformly between 0 and 1, and if  $P(\Delta E) > x$  the agent reproduces, replacing a randomly picked agent with a copy of itself.
3. With probability  $\mu$  the offspring will be mutated uniformly to another CA-rule.
4. In order to keep energy flowing into the system, after all agents have been updated, a fraction  $\gamma$  of the strings are replaced with high energy binary strings.

### Results

The main result indicates that the diversity increases as the resource level in the system drops, and this trend was investigated systematically by measuring the time average of the Shannon index shown in figure 2 and reveals that the diversity is a decreasing function of the flow and exhibits an approximately linear decrease with the flow rate  $\gamma$ , except for a saturation for high values of  $\gamma$ .

We have also studied the total population’s productivity. See figure 3. A still open question for future studies is how

tree Yggdrasil in Norse mythology. The name means well of fate.

<sup>2</sup>An online version of the platform is available at: <http://www.math.chalmers.se/~torbjrn/Urda/urda.html>

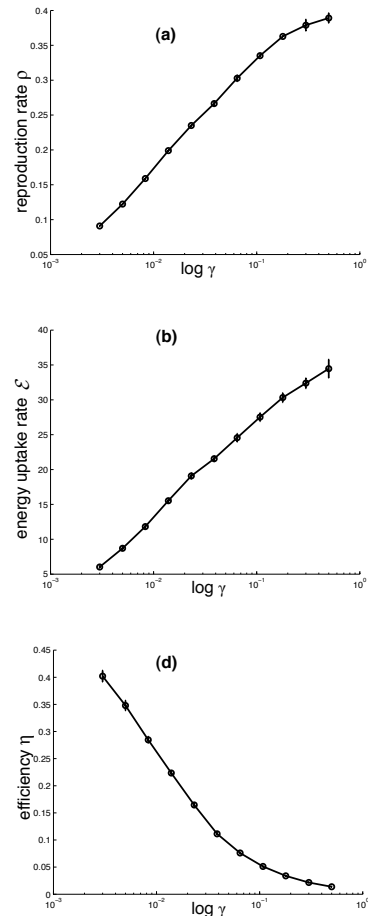


Figure 3: Three different measures related to productivity in the system. (a) shows the reproduction rate  $\rho$ , i.e. the number of divisions per update which corresponds to biomass growth, (b) shows the energy uptake rate  $\mathcal{E}$ , i.e. the energy difference between outflow and inflow, and (d) shows the efficiency of the energy uptake  $\eta$ .

good these population-productivity is compared to the optimal productivity, i.e. how good is the through evolution obtained population with respect to the total energy extraction.

### References

- Gerlee, P. and Lundh, T. (2010). Productivity and diversity in a cross-feeding population of artificial organisms. *Evolution*, 64:2716–30.
- Helling, R. B., Vargas, C. N., and Adams, J. (1987). Evolution of escherichia coli during growth in a constant environment. *Genetics*, 116:349–358.
- Waide, R., Willig, M., Steiner, C., G., M., Gough, L., Dodson, S., Juday, G., and Parmenter, R. (1999). The relationship between productivity and species richness. *Annu. Rev. Ecol. Syst.*, 30:257–300.

# On Computations and Strategies for Real and Artificial Systems

Amr S. Ghoneim<sup>1</sup>, Daryl L. Essam<sup>1</sup> and Hussein A. Abbass<sup>1</sup>

<sup>1</sup> School of Engineering and Information Technology (SEIT)  
University of New South Wales at the Australian Defence Force Academy (UNSW@ADFA)  
Canberra ACT 2600 Australia  
h.abbass@adfa.edu.au

## Abstract

A challenge in reproducing life is to reproduce cognition. We propose a methodology by which human actions are analyzed in a real-setting and are then used to evolve artificial neural networks capable of reproducing these actions. It is also demonstrated that analyzing human actions can be used for skill-assessment, where we introduce a model for in-silico computational psychology to assess skills and competency of human plays. The same methodologies can be used by coaches and mentors to diagnose skills for their players and juniors in an attempt to improve their abilities. Results demonstrate interesting patterns in the way expert players develop their skills overtime and that it is possible to reproduce these skills in an artificial context.

## Introduction

Establishing a methodology to analyze human actions has a wide spectrum of applications for ALife research. Understanding how human develop their expertise overtime can shed more light in the black-box of human intelligence. In this paper, we look at the dynamics of learning in real human, how skills develop and how the trajectory of skill-development for a human playing a complex game can be assessed. We use these findings to guide the evolution of artificial neural networks to play similarly to the human.

In an early paper in the ALife field, Stewart (1992) argued that life is cognition, that our knowledge and the way we make decisions are particularly crucial determinants for how we evolved. As was put by Varela (1995):

Yet when it comes to a re-understanding of knowledge and cognition I find that the best expression to the use for our tradition is abstract: Nothing characterizes better the units of knowledge that are deemed most natural.

Many studies focused on understanding the dynamics of learning and evolution (Floreano and Urzelai, 1998). In this paper, we analyze learning based on real human and map it to an artificial model.

In this paper, we consider the game of GO as an example where a human player needs to start from the lowest skill level, working his/her way up to establish themselves as advanced players. We needed to select a gaming environment in general and the game of GO in particular as our test platform for a number of reasons. The beauty of GO lies in the fact that: it has simple rules but large and complex search

space. First, we need an unambiguous scoring or ranking scheme. In the game of GO, this is readily available known as the system of kyu and dan ranks. Second, computer games offer a low-risk environment for prototyping artificial learning. Third, online game engines are easy and cheap sources of large amount of data. Fourth, a game such as GO is complex in its strategies, where it relies on human ability to capture spatial patterns and connect information and patterns across the whole board, an important characteristic when we design artificial games or game-theoretic models on networks. The methodology is too generic that it can be applied to both real and artificial spatial game playing.

We structure the rest of the paper in three main sections. First, we present a tiny coverage of the literature related to this paper, taking into consideration that space constraints forced us to remove many references. Second, we present the methodology and analysis using real-human players. Third, we use this analysis to evolve artificial neural networks to reproduce similar behaviours. Finally, conclusions are drawn.

## Background Material

### Skills and Competency

The term *skills* refer to the learned capacities, whether general or domain-specific, that would be crucial/useful to perform a particular job (Bassellier, et al. 2001). Skills are the component competencies that collectively create the overall *competency*; i.e., the set of skills, knowledge, and qualities or "*behavior patterns*" which are needed to allow an agent to perform tasks/ functions with proficiency (Woodruffe, 1993).

Currently, evaluating the skills of strategic board-game players depends entirely on the degree to which the game's objectives are achieved (*i.e. final outcome*). Ranking systems – whether online or offline – are virtually the only objective method for automatically assessing the players' experience. However, subjective detailed assessments can frequently be obtained from experts, where different aspects of a player's skills may be evaluated. These types of studies have traditionally been answered through psychological and skill assessment tests (Groth-Marnat, 2009). We extend this approach to a computational environment to overcome the – sometimes – instability inherent in subjective assessment and reduce the resources required to do an assessment.

## Learning

Learning can be defined as: given a *task*, a *training-experience*, and a *performance-measure*, a system is assumed to be learning “if its performance at the task improves with experience” (Thrun & Pratt, 1998). A similar model can be found in (Osherson et al., 1986), where the learning process classically requires—beside a *learner*—an *item-to-be-learned*, an *environment* wherein the learner is shown the item-to-be-learned, and finally the *hypotheses* arising to the *learner*—given the *environment*—regarding the item-to-be-learned.

This characterizes the relationship between the learning process and *experience*; a concept greatly discussed—whether explicitly or implicitly—in topics related to the “*analysis of human performance*”, or in “*studies of learning and training*” (Farrington-Darby & Wilson, 2006). Experience does not necessary lead to more powerful thinking strategies and/or acquiring directly-perceivable cues—which the inexperienced are usually aware of—but rather to a more efficient employment of the strategies and cues based on the experience-base (Klein & Hoffman, 1993). Hence, experience can “*describe skills, knowledge, or abilities, in tasks, activities, jobs, sport and games*”, and it can “*refer to a process such as decision making or [...] to an output such as a decision*” (Farrington-Darby & Wilson, 2006).

Analysing online behaviour and interaction was also investigated in (François et al., 2007), where Self-Organizing Maps were used to classify online interaction between Autistic children and robots to detect the different play styles since “*interaction is decisive in the process of learning through play*.” Also, analysing and displaying users’ activity and interaction in an online system/community, whether in a competitive way (e.g. *ranking scores*) or an non-competitive way (e.g. *activity statuses*), was found to draw users attention and motivates users participation (Deiml-Seibt et al., 2009).

## Symbiotic Adaptive Neuro-Evolution

Symbiotic Adaptive Neuro-Evolution SANE (Moriarty & Miikkulainen, 1997) is an approach to neuro-evolution where two separate populations are evolved simultaneously instead of evolving a complete network. The two populations are neurons (*explicitly decomposing the search space by acting as local solutions*) and network blueprints (*exploring the best combinations of neurons*). Blueprints are considered a better alternative to building the networks out of randomly selected neurons. Usually, SANE develop three-tiered feed-forward NNs, evolving neurons for its single hidden layer. Each neuron defines a fixed number of weighted connections that are randomly assigned to both input- and output-layer nodes.

When applied to evolve Go player, each board intersection is represented by an input node for each player, and a single output node. It is illegal to activate both nodes representing an intersection. The first input node – per intersection – is activated iff the corresponding intersection is occupied by a white stone, and vice versa if the intersection is occupied by a black stone, the second input node is activated. An empty intersection is indicated by deactivating both input nodes. A sigmoid activation function is used for the output nodes. The next move is represented by the highest value (*corresponding to the best predicted move*). If the selected move is illegal, the move corresponding to the next highest activation is selected.

However, the network passes if all its output values are below a predefined threshold (*that is, 0.5 in our experimentation*).

The evolved NNs – the blueprints population – are evaluated by playing a game(s) of Go against the selected opponent, the fitness value is merely the final score(s). As for the neurons population, the fitness value for a neuron is the normalized summation of the fitness values of the blueprints in which it participated. Single point crossover is then applied on mates selected from the elite one third of the blueprints, and 25% of the neurons, creating two offspring that replace the worst individuals. Mutation is then applied conservatively to the neuron population, and more aggressively to the blueprints (*to maintain high diversity among the network*).

## The Game of Go

The game of Go is the oldest strategic board game in the world, and is also one of the most popular. Though the game is hard, the rules of the game are few and simple, easy to learn, and flexible enough to accommodate any board size as well as the standard 19×19 board. This two-player game, where players alternate placing stones on the intersections of the board, is theoretically in the same category as Chess, as both games are intellectually stimulating, requiring high-level strategic thinking, while also giving the chance for players to apply their tactical skills (Chikun, 1997). The differences between Go and other games (including Chess) in complexity measures is obvious in (Allis, 1994), with the complexities of Go far larger than that of any of the other perfect-information games. Unlike Chess, there are no Go programs that can challenge strong human players (Van der Werf, 2004), nor even moderate human players (Ernandes, 2005). Also, although 9×9 Go boards have a complexity between that of Chess and Othello (Bouzy & Cazenave, 2001), existing Go programs are still immature.

It is worth mentioning that the best known computational model for GO is Monte Carlo Simulation. No neural network or biologically inspired model exists as yet that can outperform Monte Carlo Simulation. As such, this study is a first step to potentially take a different approach towards building neuro-players.

## Methodology

The main idea of the proposed methodology is to exploit the possible computational building block(s) of human’s actions to assess their skills and competency. The methodology estimates human’s skill and competency levels through models trained on historical data of human with known skills and competency levels. The methodology has five main steps:

**Subject Identification and Selection:** The human subjects to be selected to form the training data need to have gone through multiple competency levels. In other words, this is a longitudinal study that commences with these subjects at a low competency level then moves up to higher ones during the data collection exercise. This is the most expensive step in the whole methodology, time-wise and dollar-wise, in the real world. The game of Go traditionally uses the ranking (rating) system of kyu and dan ranks. In this paper, players with ranks ranging from 30 to 20 kyu are collectively referred to as

*Beginners*, ranks from 19 to 10 kyu are *Casual* players, 9 to 1 kyu are *Intermediate* amateur players, and finally from 1 to 7 dan are *Advanced* amateur players. Due to some ambiguities in defining the *Professional* dan ranks in the game records, we have decided not to include those ranks in the analysis. We collected the games from *No Name Go Server (NNGS)* online game-archive (Adam, 2009). The cases – game records – were selected from the years' span 1995 up till 2005. Two datasets are selected separately, a Training Dataset '*trainDS*' which is used to train the proposed classifier, and a Testing Dataset '*testDS*' from which we will select a set of Go players to observe their behaviour.

We selected 381 games for training (127 for each category; *Casual*, *Intermediate*, and *Advanced*) based on some strict rules that the games should be complete, with a registered-name, and compatible players. We did not select '*Beginner*' cases because this category contains so much noise. The reason for this noise is that it contains all players who newly joined the server, not necessarily that they are beginners but they have not played enough on this server to establish a rank.

**Data Identification and Collection:** Every action performed by the human gets recorded. In the context of a game, actions are simply the board moves. In the case of a computer board game, the state of the board at each step of the game gets saved. The training data (*i.e. the data set that will be used to build the model*) needs to be labelled (*i.e. training subjects' skills and competency levels have been assessed by some other means*), preferably with no missing values, carrying a reasonable number of records for each subject over time and that spans the subject moving from one skill level to another, and of reasonable size. The richness of the data collected per subject, as our experiments demonstrated, means that we do not require a huge dataset to build the skill-assessment model.

Four hundred games were selected for the *testDS* with only 16 games found to be common between the two datasets. The 400 games were played by 246 distinct registered-names (*i.e. players*). We imposed a threshold of at least 10 games, yielding a final set of 15 players (Table 1) to be used for testing. In the first phase of the experiments, we will run our system using the *trainDS*.

**Model Knowledge Initialization:** Skill assessment requires a richer understanding of the domain, probably more than what is needed in a traditional data mining task. What is being recorded from the interface is mostly raw data that needs to be grouped, and possibly transformed to a different representation, before it can be used properly for skill assessment. These initial features form the basis for building the actual model.

We use spatial analysis of the board to establish what we call reasons for each move. Assume a move is played in a cell, the spatial analysis will see the different shapes that are newly formed by this move. These 48 reasons are then grouped into seven categories: a category of what it seems a bad move (*anti-suji*), a category for attack, a category for defence, a category for gaining an advantage, a category for deep planning, a category for end of game and an overall category of all reasons put together. These seven categories are named as: "*Not Recommended*", "*Considered an Attack*", "*Considered A Defence*", "*Explicit Gains*", "*Thoughtful*", "*End of the Game*" and "*All Reasons*" respectively.

Player ID	Number of Games Played	The Averaged Experience Range Covered by the Corresponding Player's Games
1	74	Upper-Beginner to Lower-Intermediate
2	38	Lower-Intermediate to Mid-Intermediate
3	32	Mid-Intermediate
4	46	Mid-Beginner to Mid-Casual
5	20	Lower-Intermediate
6	16	Mid-Casual to Lower-Intermediate
7	35	Mid-Casual to Upper-Casual
8	13	Mid-Casual to Upper-Casual
9	36	Upper-Intermediate
10	26	Lower-Intermediate to Mid-Intermediate
11	50	Lower-Advanced
12	11	Lower-Advanced
13	57	Upper-Intermediate to Lower-Advanced
14	10	Mid-Intermediate
15	34	Lower-Intermediate to Mid-Intermediate

Table 1: The final test dataset

It is obvious from the plain definition of each category that these categories can overlap. The Frequencies '*F*', Frequencies per Step '*FS*', and the Percentages '*P*', are applied as measurements for the aggregated subsets of the generated-reasons per game. Subsequently, and between each distinct pair of experiences, the Wilcoxon-test and a two-sample T-test were applied to statistically signify the ability of the calculated medians/means to differentiate between the corresponding distinct pair of experiences.

**Model Building:** The model can vary in its characteristics, ranging from simple statistics to complicated neural networks, decision trees, or classifier systems. The choice of the features in the previous step and the right model in this step are critical and can create all the differences between good or bad skill assessment models.

The *Median* and the *Median Absolute Deviation (MAD)* (Davies & Gather, 1993) were chosen as robust univariate measures in case the dataset is contaminated by outliers (*i.e. observations which appear to be inconsistent with the remainder of the dataset*) and thus subject to masking and/or swamping effects. Human players can be of a wide range of experiences, spanning from beginners to professionals. Given the set of experiences  $E = \{e_1, e_2, \dots, e_n\}$ , let  $D_e$  denote a subset of the dataset of all games  $D$  where the experience of both opponents is  $e$ . The median can be estimated as:

$$Median_{e,s,\varphi} = \left( \varphi_{[(|D_e|+1)/2]:|D_e|,R_s} + \varphi_{[|D_e|/2]+1:|D_e|,R_s} \right) / 2$$

where  $\varphi$  is the measurement function (*i.e. denoting F, FS, and P*),  $R_s$  is the *sth* reasons subset,  $|D_e|$  is the number of games in  $D_e$ , and  $\varphi_{1:|D_e|} \dots \varphi_{|D_e|:|D_e|}$  are the order statistics of

$\varphi_1 \dots \varphi_{|D_e|}$ . Accordingly, MAD can be estimated as:

$$MAD_{e,s,\varphi} = Median \left( \left| \varphi_{1:|D_e|,R_s} - Median_{e,s,\varphi} \right|, \dots, \left| \varphi_{|D_e|:|D_e|,R_s} - Median_{e,s,\varphi} \right| \right)$$

The medians of the different reasons subsets can model how the general strategy is decomposed into characterizing sub-



Measurements	Reasons' Subsets	Casual Games		Intermediate Games		Advanced Games	
		Median	MAD	Median	MAD	Median	MAD
Frequencies (F)	Not Recommended	28	8	35	9	34	7
	Considered An Attack	262	80	337	112	357	84
	Considered A Defence	400	79	471	84	484	72
	Explicit Gains	123	13	138	13	139	14
	Thoughtful	134	40	175	47	189	44
	End of the Game	0	0	1	1	2	1
	All Reasons	840	174	1021	191	1053	163
Percentages (P)	Not Recommended	3.493450	0.5431392	3.217822	0.4167558	3.222919	0.3764555
	Considered An Attack	30.89655	3.544815	33.26510	3.895654	33.63148	2.681764
	Considered A Defence	46.64372	1.643718	45.64995	1.862967	45.76271	1.503475
	Explicit Gains	15.23702	3.438733	13.83588	3.526759	13.05903	2.960512
	Thoughtful	15.57943	1.648398	16.92677	1.553367	17.68140	1.462031
	End of the Game	0	0	0.1154734	0.1154734	0.1552795	0.09492900
Frequencies Per Step (FS)	Not Recommended	0.1206226	0.02429229	0.1275862	0.02576802	0.1269231	0.02250541
	Considered An Attack	1.039024	0.2743185	1.261649	0.3359982	1.322222	0.2477437
	Considered A Defence	1.628099	0.1956418	1.730375	0.2213487	1.801394	0.1727017
	Explicit Gains	0.5088968	0.06054765	0.5152838	0.06837375	0.5154639	0.06208771
	Thoughtful	0.5527273	0.1229400	0.6518518	0.1434485	0.7003484	0.1255453
	End of the Game	0	0	0.00387596	0.00387596	0.00666666	0.00361788
	All Reasons	3.418118	0.4618815	3.810169	0.5740072	3.941176	0.4627970

Table 2: The Medians and Median Absolute Deviations (MAD) of the different subsets, among diverse experiences

strategies, and demonstrates the variations in the strategies employed by human Go players of different experiences. To confirm the potential hypotheses suggested by the data, both a two-sample T-test and a two-sided Wilcoxon rank sum test are used. By permuting reasons subsets, estimated measurements, and pairs of different experiences, the T-test and Wilcoxon-test will respectively examine the null hypothesis that the data – *measurements per game* – have equal means/medians against the alternative that the means/medians are not equal.

The two-sample T-test tests a null hypothesis  $H_0$  that the two independent samples come from normal distributions with unknown variances and the same mean, against the alternative that the means are unequal. The test is two-tailed, and performed at a significance level  $\alpha = 0.05$ , i.e. the probability of mistakenly rejecting  $H_0$  (*Type I error*) is no more than 5%. Alternatively, the Wilcoxon-test tests a null hypothesis  $H_0$  that the two independent samples come from identical continuous distributions with the same median, against the alternative that the medians are unequal. The Wilcoxon-test is also performed at  $\alpha = 0.05$ .

In this study, a three-tier ensemble is used to predict the class label of a game of Go as *Casual*, *Intermediate*, or *Advanced*. The first-tier is based on Random Forests (RFs) (Breiman, 2001); ensembles of Classification Decision Trees (CDTs). In order to analyze the reasons, we are looking for a robust white-box model, which can handle data without requiring a lot of data preparation. These requirements suggest the use of CDTs. Each individual classifier (i.e., *RF*) is trained to classify a class and its complement; for example, a RF is trained to classify *Casual* games versus *Not-Casual* (i.e., *Intermediate* and *Advanced*) games, and so on. Thus, each RF outputs two probabilities 'Pr'; i.e. for the previous example, a probability that a given game is originating from the *Casual* class:  $\text{Pr}(C)$ , and a probability that the same given game is originating from the *Not-Casual* class:  $\text{Pr}(\neg C)$ .

In our experiments, a RF is an ensemble of – a maximum of – 1000 classification decision trees. A forest's attributes are

determined according to the *Error* and the *Size*; respectively, the minimum error (i.e., *misclassification probability for the out-of-bag observations*) recorded during the process of adding up trees while creating the forest and the ensemble size (i.e., *number of trees*) corresponding to that error value.

The second-tier creates an ensemble of RFs (i.e., a *Forest of Random Forests*) for each class then the joint probability distribution is calculated for two cases: that the instance belongs to the class and that the instance does not belong to the class. The third and final tier combines the results from the second-tier forests using a final gate-function to create for each observation (i.e., *game*) a single probability 'Pr<sub>final</sub>' per class. The final gate-function combines the probability that an instance is from one class and the probabilities that this instance does not belong to other classes.

Table 2 shows the medians and median absolute deviations among the 127 games per skill level and reasons' subsets. Table 2 shows a statistical difference between casual/intermediate and casual/advanced human Go players, yet it fails to differentiate between intermediate/advanced players. The medians tend to get higher with experience considering both *F* and *FS* as measurements; the only exceptions are when the medians of the advanced are lower than or almost equal to the corresponding intermediate in both subsets *Not Recommended* and *Explicit Gains*. Though this apparent correlation between the *F/FS* medians and the growing experience is expected to some extent, because more experienced players tend to play longer games, the two previously mentioned cases highlights the possibility that more experienced human players are less attracted by direct/instant gains and are more considerate when it comes to not recommended moves. This possibility is supported by medians reported for the measurement *P*, where the medians of both subsets almost decrease with growing experience.

Using *P* again, the medians of the subset *Considered A Defence* somewhat decreases with growing experience, suggesting that a more aggressive strategy is applied by well-

experienced human players, as opposed to a more defensive strategy by their less-experienced counterparts. The later suggestion is supported by the medians reported for the subset *Considered An Attack* which increase in correlation with growing experience in view of all of the three measurements. The medians reported for both subsets *Thoughtful* and *End of the Game* also increase in correlation with growing experience in view of all measurements.

Thus we can generally claim that, with rising experience, a human player's strategy evolves to a more thoughtful and aggressive strategy, a strategy that cares more about the final steps and eludes the not recommended moves, and last but not least, a strategy that is less lured by direct gains. This claim is statistically supported for human players who progress from casual to both intermediate and advanced experiences.

While using reasons greatly simplified and abstracted the typical knowledge used by humans, the use of aggregated sets of reasons additionally shortened the available reasons and allowed for the highest possible level of strategic abstraction. The three proposed measurements proved to be reasonable in quantifying the strategical aspects of the varying experiences.

Using the features generated per game, an initial preprocessing step is carried on by applying the Minimum Covariance Determinant (MCD) algorithm for outliers' detection. A MCD  $\alpha$ -value of 0.7 was selected, and all the games tagged as outliers were excluded from the *trainDS*. In this study, outliers are not considered noise or error, rather they are assumed to carry important information that accounts particularly for any unaccounted for parameters when selecting the dataset (for example, the length – number of moves per game). This step is followed by growing RFs that aim to use the previously calculated features to classify the games according to players' ranks. The preprocessing step showed that the measurement *FS* appears less affected by the potentially different or varying mechanism responsible for the outliers. Thus, *FS* is selected as the reliable measurement to monitor the players' competency and skills.

Using the uncontaminated *trainDS*, 30 RFs are trained to differentiate between each experience level and its complement. The 30 RFs trained – per experience, and using the *FS* – are combined to form the second-tier ensemble.

**Model Testing:** Once the model is built, it gets tested with subjects that were not included in the model building exercise. Upon successful testing, the model is ready for use.

Using the *testDS*, the games for each player are temporally ordered and then reasons were extracted to estimate the strategic reasoning behind the moves. *FS* is then applied – and combined according to the aggregated reasons' subsets – thus creating the final feature set for each game. The proposed classifier generates three final probabilities for each game:  $Pr_{final}(C)$ ,  $Pr_{final}(I)$ , and  $Pr_{final}(A)$  for Casual, Intermediate and Advanced respectively. Given the number of available experience-levels as  $N_{classes}$ , and the total number of games per a single player as  $N_{games}$ , three Competency Monitoring-Curves are plotted for each player; each representing the 'un-weighted' Cumulative Moving Average (CMA) for an experience-level, with a maximum window size of 50 games.

For space limitations, we will only show the results for one player with average predictive results to make the discussion more interesting. Figure 1 presents a 2-dimensional line graph

with two y-axes for the player. The left y-axis – 'Player's Experience Curves' – displays the value of the three Monitoring-Curves (i.e., generated probabilities), while the right y-axis – 'Ranks' Categories' – displays the Player's Rank according to the online NNGS archives. The Player's Rank curve is also a CMA of the actual rank-values. A label on the right y-axis represents the center of the respective rank category. The x-axis displays the game number, with imposed temporal frames for the corresponding dates (months/years).

The monitoring-curves in all of the resulting figures (including those not-shown) show a clear consistency between the experience level of a player and his/her probabilities' curves. That is, as the player 'assumably' gains more experience with time, the probabilities' curves reflect this learning activity by either declining or rising. Player#1 advances from an Upper-Beginner to a Lower-Intermediate experience over the course of the 74 games selected. Concurrently, the Casual-monitoring-curve of the mentioned player declines from 0.875 to around 0.5625, the Intermediate-curve also converges to around 0.5625 rising from 0.4375, and the Advanced curve is also rising from 0.3750 to a little bit higher than 0.5.

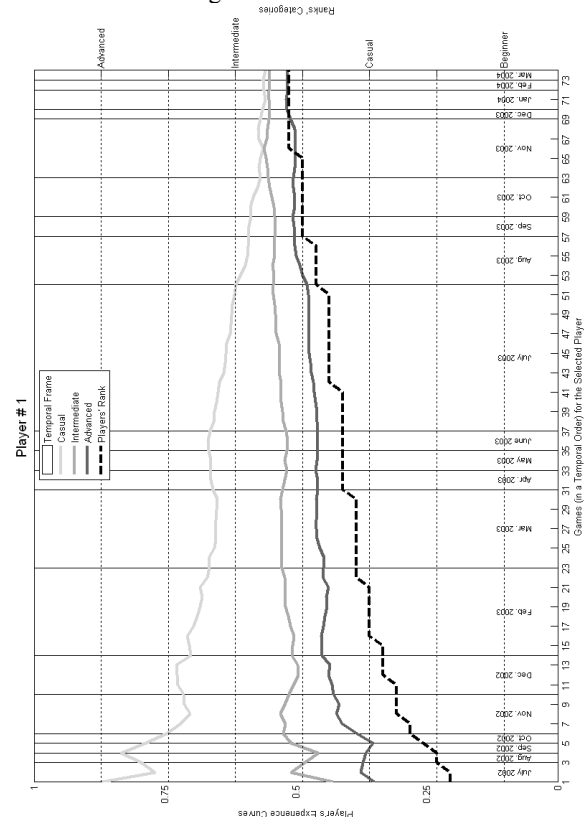


Figure 1: The Competency Probability-Curve for Player 1

Though, on strict classification bases, this player is classified as a Casual during the whole period (since the Casual probability curve is higher than both the Intermediate and Advanced), the clear trend in the curves assures that – with more games – the player is going to be correctly classified as Intermediate. Obviously, classifying a Beginner player as a Casual is reasonable in this context, since no Beginner cases were included in *trainDS*.

Here we reach the final stage of our results, in which we diagnose the skills learning-activity of human Go-players by temporally observing each of the strategies’ characteristics. A straight benefit of the figure is the distinctive opportunity to realize how the strategic reasoning of human Go-players is decomposed among the available strategies’ characteristics, and how those characteristics evolve temporally with experience. Figure 2 shows the normalized un-weighted CMA of the *FS* directly measured from the games. As a player progresses from being a Beginner to lower-Intermediate – through being a Casual – the categories ‘All Reasons’, ‘Considered an Attack’, ‘Considered a Defense’, and ‘Thoughtful’ seem – in general – to decline slightly with experience. The ‘Explicit Gains’ appears to be the only curve rising during the Beginner to lower-Intermediate progression. These findings are obvious in Figure 2. On the contrary, progressing from the Intermediate to Advanced shows precisely the opposite behavior. As players progress through

Our earlier findings seem to agree with only half of the later findings, that is, the changes occurring as a player progresses through the Intermediate rank and to being an Advanced. This apparent disagreement – where the categories ‘All Reasons’, ‘Considered an Attack’, ‘Considered a Defense’, and ‘Thoughtful’ seem to decline as a player advances from being a Beginner to a lower-Intermediate – can be attributed to two associated reasons. At first, in Table 2 no Beginner games matched the selection criteria, and therefore, the progression from Beginner to Casual was not investigated. That leads us to the second reason; the window size of 50 games employed in the CMA considers this ‘history’ of being a Beginner when the player has already advanced onto being a Casual, thus affecting the curves for an additional period.

## Strategically Aware Fitness Measurement

To accomplish this goal, SANE was used to evolve  $9 \times 9$  neuro-Go players using both the traditional exclusively score-dependent fitness function, and the proposed strategically-aware function. The networks are evolved against the GNU Go engine as an opponent. To compensate for the additional computational cost of estimating the strategies in the Go games, 50 blueprints are evolved instead of the 200 suggested by (Richards, et al. 1998). Due to the nature of the problem in which a network is evaluated by playing a game, and in spite of using elitism, the fitness values across generations fluctuate due to stochastic effect.

$$f_{Network} = \sum_{i=1}^{N_{Games}} \frac{\alpha Score_i + (1-\alpha)TP_i}{N_{Games}}$$

265

representing the *Network's* score in game  $i$ , while  $TP_i$  is the *Trained Probability* generated by the RF for the game  $i$ .

To investigate the effect of the added TP, the coefficient  $\alpha$  was varied, using four values;  $1.0$ ,  $0.8$ ,  $0.2$ , and  $0.0$ . The first  $\alpha$  value represent the traditional score-based fitness function, while  $\alpha$  set to  $0$  represents the case where the networks' evaluation is based entirely on the Trained Probability.

In General, the parameters in the experiments are based on those found effective in (Richards, et al. 1998), except for the number of blueprints which was reduced from  $200$  to  $50$ . A single run consists of  $500$  generations, and  $10$  different runs were evolved. The  $500$  generations are twice the number of generations required by SANE to evolve a network capable of defeating Wally on  $9 \times 9$  boards in (Richards, et al. 1998). Since Wally is a trivial engine when compared to GNU Go, GNU Go's level was set to  $1$  throughout the experiments instead of the default of  $10$ . However, GNU Go – even when playing at level  $1$  – is much more developed than Wally. Therefore, we do not expect to evolve a NN that is capable of defeating GNU Go, but a network that has developed enough strategies to be explored.

The games were scored using Chinese rules. The networks were always evolved to play White, thus never making the first move. The komi value – necessary to avoid a tie – was set to  $0.5$ , and no handicap stones were given to the networks. An upper bound of  $200$  moves per game was placed, to ensure that unreasonably long move-sequences that are probably suggested by the untrained networks are not pursued. This experimental setup cost up to a maximum of  $27$  days for a single run per an  $\alpha$  value using a Sun Constellation Cluster.

## Results and Discussion

In order to investigate the effect of the proposed fitness function, two different types analysis – to the neuro-evolution process – are to be shown and discussed. We start by showing and discussing the convergence among the varying  $\alpha$  values. Then a Tournament between selected players and GNU Go – set to different levels – is held.

Figure 3 shows the convergence of the  $50$  blueprints and  $4000$  neurons evolved using SANE for  $500$  generations. For each  $\alpha$  value, the average fitness – of  $10$  different runs – for the 1) best network, and the 2) entire population are plotted.

The convergence of the fitness values of all of the different combinations enters a relative plateau, starting from around generation number  $50$  for both  $\alpha$  values of  $0.2$  and  $0.8$ , and followed by generation number  $150$  for both  $\alpha$  values of  $0.0$  and  $1.0$ . The same is true for the convergence of the entire population, except for  $\alpha = 0.0$  where the population seems to continue evolving. Notably, the 'relative' difference between the best network and the population in terms of fitness values decreases with an increasing  $\alpha$ , except for  $\alpha = 0.8$  which shows the lowest difference. A possible explanation is that while depending more on the TP component rather than the score, the evolving networks increasingly fluctuate between the generations. However, setting to  $\alpha = 0.8$  shows a less varying fitness than  $\alpha = 1.0$ , even in other detailed figures that are not shown here due to page constraints.

The first step to investigate the playing capabilities evolved and whether it takes advantage of the engine's weaknesses is

by holding a tournament between selected representative players and GNU Go. As we mentioned before, SANE used a weaker player than GNU Go at level  $1$  as an opponent. The behavior of the evolved players when playing against GNU Go set to different levels will shed a light on the type of the strategies evolved. The tournament involves GNU Go at  $10$  different levels, starting from  $1$  (weakest) to  $10$  (default).

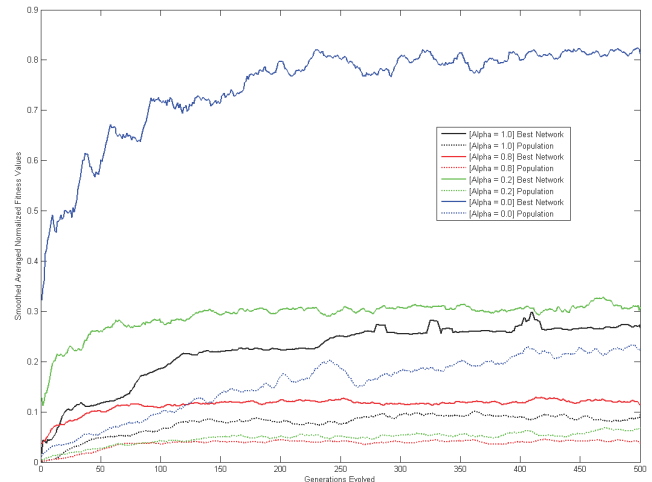


Figure 3: The Convergence of the Fitness Values

A simple and straightforward criteria is used to select a representative player for each of the varying  $\alpha$  values; the network achieving the overall best 'game score' across the  $10$  runs and the  $500$  generations. Figure 4 shows the best games' scores across the  $10$  runs, the best score for each  $\alpha$  is encircled. The maximum possible score using the Chinese rules – and a komi value of  $0.5$  – on a  $9 \times 9$  board is  $81.5$ .

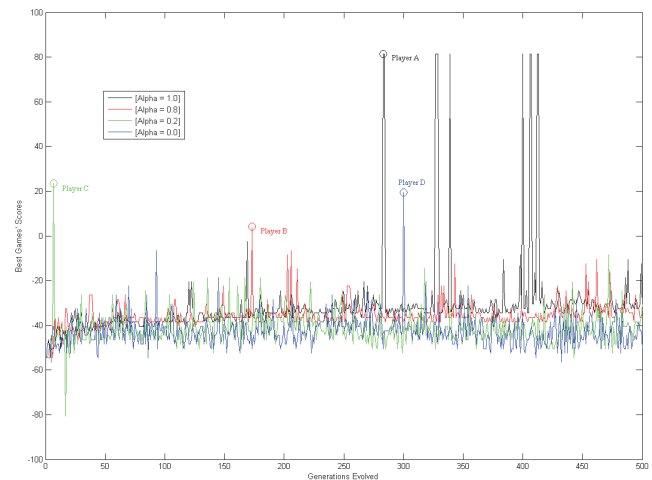


Figure 4: Best Games' Scores across the 10 runs

The tournament consists of the four selected players versus GNU Go at  $10$  different levels. The players will be named *PlayerA*, *PlayerB*, *PlayerC*, and *PlayerD*; representing respectively the  $\alpha$  values of  $1.0$ ,  $0.8$ ,  $0.2$ , and  $0.0$ . For each pair – that is, a selected player versus a GNU Go at a single level –  $30$  different matches were played. The komi value is set to  $0.5$ , the games are scored using Chinese rules, and the GNU Go always starts the games.

Table 3 shows the percentage of Wins of the 4 selected players against GNU Go. Even though none of the players were able to defeat GNU Go at a level higher than the one they were evolved against, as  $\alpha$  decreases, the percentages of wins against GNU Go at level 1 increases until it reaches 20% of the games for *PlayerD*. This finding strongly suggests that the networks evolved using the proposed fitness function evolve different varying strategies to defeat the opponent. Even if those players were selected from premature generations; *PlayerC* was evolved in the seventh generation.

		Selected Players			
Details	<i>Alpha Value</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
	<i>Corresponding Generation</i>	283	173	7	300
The Level of GNU Go	1	6.7%	10%	13.3%	20%
	2	0%	0%	0%	0%
	3	0%	0%	0%	0%
	.	.	.	.	.
	.	.	.	.	.
	10	0%	0%	0%	0%

Table 3: Selected Players' Details and Percentages of Wins

The main objective in a game of Go is to secure a territory. The capability of creating and defending a group of connected stones that remains alive – i.e., do not get captured – until the end of the game is fundamental to a go player. Therefore, the final scores of the games, even in cases of losing, are meaningful to our analysis. Players that can secure bigger territories than other players, and which will be reflected in the final score, are relatively more trained.

Figure 5 shows the average score of the selected players against GNU Go. Since GNU Go always plays as the black, and given the komi value of 0.5, the minimum possible score for the selected players is -80.5. All players report their best results when playing against level 1. For higher levels, Players C and D report the minimum possible score. However, *PlayerB* reports better average scores in most of the higher levels than *PlayerA*.

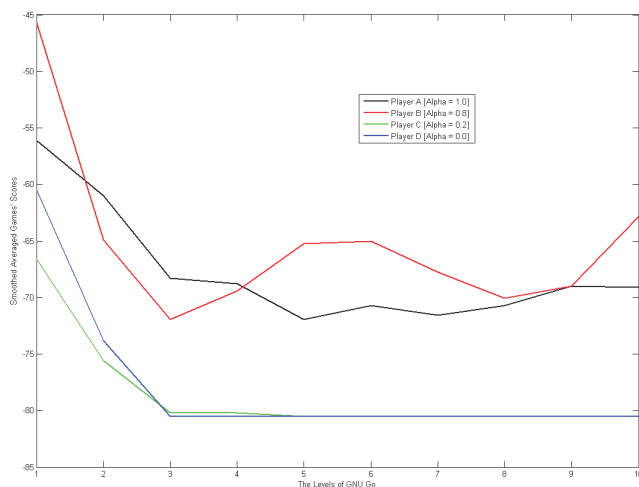


Figure 5: The Average Score against GNU Go

## Conclusions and Future Work

We provided a methodology for an automatic and objective assessment and monitoring of human-players' skills and competencies in the game of Go. The generality of the approach entails that the models can be used to assess artificial players as well, which we successfully demonstrated using an Artificial Neural Network. The findings are seen as advancement towards better understanding of human strategies to assess the skill levels of humans. For example, if player's skills are constant for a while, and if the objective is to improve the performance of that player, the artificial life environment or a game environment may switch to some training scenarios to improve the specific skills which have been stagnating. If the aim is to entertain the person, the game may alternate between an easier version and a harder version.

## References

- Adam, G. (2009) No Name Go Server Archives. <http://computer-go.org/pipermail/computer-go/2006-April/005343.html> (12-Oct-09)
- Allis, L. (1994). *Searching for Solutions in Games and Artificial Intelligence*. Ph.D. thesis, Maastricht University, The Netherlands.
- Bassellier, G., Reich, B. and Benbasat, I. (2001). Information Technology Competence of Business Managers: A Definition and Research Model. *Management Information Systems, J.*, 17:159–182.
- Bouzy, B. and Cazenave, T. (2001). Computer Go: An AI oriented survey. *Artificial Intelligence*, 132:39–103.
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45:5–32.
- Chikun, C. (1997). *Go: A Complete Introduction to the Game*. Kiseido.
- Davies, L. and Gather, U. (1993). The Identification of Multiple Outliers. *American Statistical Association, J.*, 88:782–792.
- Deiml-Seibt, T., Pschetz, L. and Müller, B. (2009). A conversational model to display user activity. In *23rd BCS HCI 09 Conf. on People and Computers*, Swinton, UK.
- Ernandes, M. (2005). Artificial Intelligence & Games: Should Computational Psychology be Revalued? *Topoi*, 24:229–242.
- Farrington-Darby, T. and Wilson, J. (2006). The nature of expertise: A review. *Applied Ergonomics*, 37:17–32.
- Floreano, D. and J. Urzelai (1998). Evolution and learning in autonomous mobile robots. In D. Mange and M. Tomassini (Eds.), *Bio-Inspired Computing Machines: Towards Novel Computational Architectures (1st ed.)*, Chapter 12, pages 317–364.
- François, D., Polani, D., and Dautenhahn K. (2007). On-line behavior classification and adaptation to human-robot interaction styles. In *ACM/IEEE Inter. Conf. on Human-robot interaction*, NY, USA.
- Groth-Marnat, G. (2009). *Handbook of psychological assessment*. Wiley.
- Klein, G. and Hoffman, R. (1993). Seeing the Invisible: Perceptual-Cognitive Aspects of Expertise. In Rabinowitz, M. (ed), *Cognitive Science Foundations of Instruction*. Lawrence.
- Moriarty, D., and Miikkulainen, R. (1997). Forming Neural Networks through Efficient and Adaptive Coevolution. *Evolutionary Computation* 5, 4:373–399 Erlbaum Associates, New Jersey.
- Richards, N., Moriarty, D., and Miikkulainen, R. (1998). Evolving Neural Networks to Play Go. *Applied Intelligence*, 8:85–96
- Osherson, D., Stob, M. and Weinstein, S. (1986). *Systems that learn*. MIT Press Cambridge, Mass., USA.
- Stewart, J. (1992). Life=Cognition The epistemological and ontological significance of Artificial Life. In F. Varela and P. Bourgine: Toward a Practice of Autonomous Sys., *Euro. Conf. on A. Life*, MIT Press.
- Thrun, S. and Pratt, L. (1998). *Learning to learn*. Kluwer Academic Pub.
- Van der Werf, E. (2004). *AI Techniques for the Game of Go*. Ph.D. thesis, Universiteit Maastricht, Maastricht, The Netherlands.
- Varela, F. J. (1995). Some Biological Ingredients for a Nouvelle Cognitive Science. In L. Steels and R. Brooks (Eds), *The Artificial Life Route to Artificial Intelligence*, 11–22.
- Woodruffe, C. (1993). What is meant by a competency? *Leadership & Organization Development, J.*, 14:29–36.

# Non-Template Molecules Designed For Open-Ended Evolution

Chris Gordon-Smith<sup>1</sup>

<sup>1</sup>SimSoup  
c.gordonsmith@gmail.com

## Abstract

Theories of the Origin of Life can be categorised as ‘template replication first’ and ‘metabolism first’. A key question for metabolism first theories is whether metabolic systems can support open-ended evolution; this is related to the number of possible persistent states of such a system. Earlier work<sup>1</sup> has demonstrated that artificial chemical systems can have memory; an essential requirement for inheritance. The current paper extends this, taking a ‘proof of concept’ approach to the question of the number of persistent states. It shows an artificial chemical network forming a ‘memory bank’ with many possible states. It also makes the link between chemical network structure and molecular structure, and provides a design for a set of artificial molecular species for the memory bank network. Preliminary simulation results from the SimSoup artificial chemistry simulator are included, confirming the operation of an initial set of ‘memory units’. The work supports the view that open-ended evolution can begin without requiring highly complex template molecules.

## Motivation, Approach, And Paper Overview

Metabolic theories of the Origin of Life propose that early organisms were metabolic systems that transmitted inherited information without the use of template replicating molecules such as DNA and RNA, and without the very complex mechanisms needed for their accurate replication<sup>2</sup>.

It is envisaged that the systems were individuals capable of growth and reproduction; in some theories they are thought of as protocells that could divide. Variations in the metabolisms of different individuals would have led to differences in fitness that would drive evolution.

For this to be workable, successful variations would have to be retained and passed on to offspring. In addition, for evolution to be effective it would need to be open-ended, with a large number of possible variations in metabolism.

The motivation for this paper is to investigate whether this is feasible. A ‘proof of concept’ approach is adopted in

<sup>1</sup>See Gordon-Smith (2005, 2007, 2009a,b) for earlier papers including SimSoup model details, and SimSoup (2011) for open source program code.

<sup>2</sup>Such mechanisms are prebiotically implausible, and so problematic for template replication first theories.

which an artificial chemical network and associated molecular structures are *designed* for open-ended evolution. If the structures identified are not too complex, then it is reasonable to suppose that molecules with similar capabilities and properties could have occurred in the prebiotic world.

The rest of this paper includes the following:<sup>3</sup>

- *Conceptual Background* inspiring this work
- *Memory In Chemical Networks*:
  - *A Network Oriented View Of Chemistry*: A description of the *Network Components* from which chemical networks are constructed, the way these can be combined to form more complex *Compound Interactions*, an explanation of the distinction between *Static and Dynamic Networks*, and a discussion of *Catalysis* from a network point of view
  - *Network Memory And Exploration*: A description of a network that forms a *Two State Memory Unit*, and a discussion of how such units can be put together such that *The Dynamic Network Explores The Static Network*
- *Network Structure For High Memory Capacity*: A description of a network in which many *Memory Unit Sub-Networks* are combined to form a *Memory Bank Network*
- *Molecular Structure For The Memory Units*: A detailed description of a set of molecular structures in the SimSoup artificial chemistry that have been designed to implement the Memory Bank Network. This section describes:
  - *Molecular Structure In SimSoup*
  - *Atom Types For The Memory Unit Molecules*
  - *The Memory Unit Molecule Structures And Dimer Splitting*: Structures of Memory Unit Molecules, and of Dimer Splitting that is key to its operation
- *Results* of ‘proof of concept’ tests to investigate the workability of the memory bank
- *Discussion, Conclusions And Prospects, and References.*

<sup>3</sup>Section names are italicised.

## Conceptual Background

The SimSoup project takes inspiration from:-

- Metabolic theories including those of Aleksandr Oparin (Oparin, 1957), Stuart Kauffman (Kauffman, 1993), Freeman Dyson (Dyson, 1999), Chrisantha Fernando and Jonathan Rowe (Fernando and Rowe, 2007), and the Lipid World theory and GARD model of Doron Lancet's group (Segré et al., 1998, 2001a,b)
- Graham Cairns-Smith's clay crystal and genetic takeover theory (Cairns-Smith, 1982)
- Tibor Gánti's work on the principles of life and chemoton theory (Gánti, 2003)
- Network theory, particularly the work of Sanjay Jain and Sandeep Krishna (Jain and Krishna, 1998; Krishna, 2003)
- The Chemical Organisation Theory of Peter Dittrich and Pietro Speroni di Fenzio (Dittrich and di Fenzio, 2007)
- Günter Wächtershäuser's chemo-autotrophic Iron-Sulphur World (Wächtershäuser, 1990, 1997, 2006)
- Linus Pauling's chemical bond theory (Pauling, 1960).

## Memory In Chemical Networks

### A Network Oriented View Of Chemistry

This section presents a network oriented view of chemistry, and introduces terminology used in SimSoup.

**Network Components** The basic units of chemistry are particles and elementary reactions between these particles. The particles can be molecules or ions and are of different types (species). In an elementary reaction, one or more particles reacts directly to form products in a single reaction step and with a single transition state.

In SimSoup, a species of particle is called a *Molecule Type*, and an elementary reaction with particular Reactant(s) and Product(s) is called an *Interaction Type*. An instance of a Molecule Type is a *Molecule*, and an instance of an Interaction Type is an *Interaction*.

From a network point of view<sup>4</sup>, there are only three forms

<sup>4</sup>A network constructed from elements as shown in Figure 1 is not a graph in which the vertices represent Molecule Types and the edges represent Interaction Types. Constructions and Fissions each have three vertices connected by two edges, whereas each edge in a graph has only two vertices.

A chemical network can be represented by a *directed bipartite graph*. A bipartite graph has vertices that can be divided into two disjoint sets U and V such that every edge connects a vertex in U to one in V. In a directed bipartite graph, each edge has a direction.

Alternatively, a chemical network can be represented by a *directed hypergraph*; a hypergraph is a generalisation of a graph in which a 'hyperedge' can connect any number of vertices. In a directed hypergraph, hyperedges connect 'head' vertices to 'tail' vertices. The network elements of Figure 1 can be regarded as edges in a directed hypergraph.

of elementary reaction as follows (see Figure 1):-

- **Construction:** Two Reactant Molecules join to form a single Product Molecule
- **Transformation:** A single Reactant Molecule re-arranges to form a Product with the same atomic composition, but different structure
- **Fission:** A single Reactant Molecule splits to form two Product Molecules.

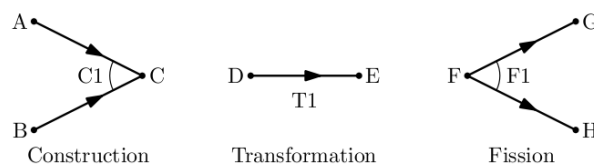


Figure 1: The three forms of Interaction Type. In Construction C1, Reactant Molecules of types A and B join to form a Product of type C. In Transformation T1, a Molecule of type D re-arranges to form a Molecule of type E. In Fission F1, a Molecule of type F splits into Molecules of types G and H.

**Compound Interactions** More complex reactions can take place as a result of Interaction Types combining in various sequences. Figure 2 shows a compound interaction with overall scheme  $A + B \rightarrow E + F$ .

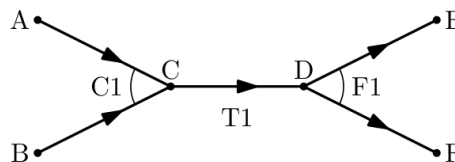


Figure 2: A Compound Interaction

A Compound Interaction does not have a rate constant that determines the reaction rate according to the concentration(s) of the (non-intermediate) Reactant(s). In Figure 2, the reaction dynamics depend on the concentrations of C and D, as well as of A and B. If the Compound Interaction forms part of a larger network, C and D may be Reactants or Products for other Interaction Types, and so the reaction depends on factors other than the concentrations of A and B.

**Static And Dynamic Chemical Networks** A set of Molecule Types and Interaction Types (along with temperature and pressure dependent rate constants) define a *static network*. This is determined for all time by the laws of physics. A *dynamic network* is a set of actual Molecules and actual Interactions taking place between them at particular rates. As such, it is a possible process that can occur within the framework of a static network.



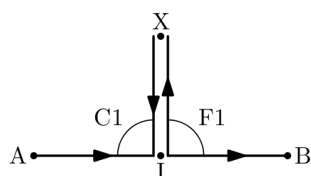


Figure 3: Catalysis Example

**Catalysis** The word ‘Catalyst’ does not denote a kind of Molecule. It denotes a *role* that a Molecule can play in a chemical process. In Figure 3, X plays the role of a catalyst; it is used by Construction C1, and released by Fission F1, so that overall it is neither consumed nor produced.

## Network Memory And Exploration

**A Two State Memory Unit** Figure 4 shows a simple (static) network for an artificial chemistry consisting of three elementary reactions C1, F1 and F2:

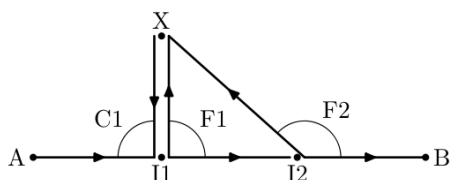
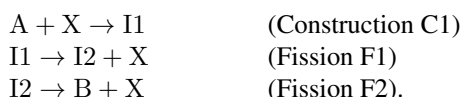


Figure 4: A two state chemical memory unit.

A is abundantly available ‘food’; initially no other Molecules are present. In the absence of X Molecules, Construction C1 cannot proceed and A remains the only Molecule Type present. If a single Molecule of X is introduced, a Molecule of I1 is produced (Construction C1). This then splits (Fission F1) to release an X Molecule and an I2 Molecule. The I2 Molecule then splits (Fission F2) to release another X Molecule plus a B Molecule. Overall, for each A Molecule consumed, one new X Molecule becomes available in addition to the B Molecule. As a result, the supply of X is maintained (even if there is some ‘leakage’).

The network is bistable; it has two states, one in which only A Molecules are present and no Interactions occur, and another in which Interactions proceed and X is maintained. The introduction of a single Molecule of X is ‘remembered’ because it triggers a switch to a new persistent state.

The network therefore constitutes a simple memory unit with an information capacity of 1 bit<sup>5</sup>.

<sup>5</sup>Under the current design for the memory unit, state changes of the unit are not reversible. However, such changes *can* be reversed at the ecosystem level. See the ‘Discussion’ section below.

## The Dynamic Network Explores The Static Network

Figure 5 shows a static network in which two of the memory units in Figure 4 are connected in series.

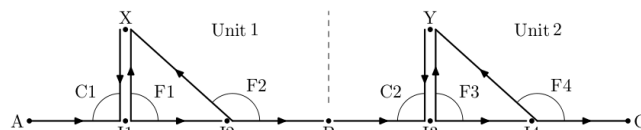


Figure 5: A two unit memory network with three states

If only A is available as ‘food’, there are three possible persistent states of the dynamic network: i) neither unit is active (only A is present), ii) only unit 1 is active, iii) both units are active.

In a more general situation where the static network is (effectively) infinite, we can consider a dynamic network to be ‘exploring’ the static network. A perturbation (such as the addition of a single X or Y molecule) can cause new parts of the network to become accessible.

## Network Structure For High Memory Capacity

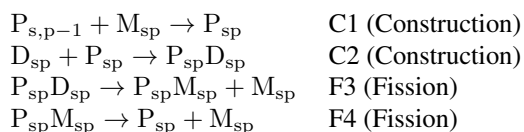
The previous section described how simple two-state memory units can be combined to form a larger network with more stable states and so higher memory capacity.

This section presents a network that systematically combines a large number of memory units to form a network with a correspondingly large memory capacity.

## Memory Unit Sub-Network

Figure 6 shows a two state network that will form a memory unit within a larger ‘memory bank’ network<sup>6</sup>.

$P_{s,p-1}$  and  $D_{sp}$  are ‘food’. The Interaction Types in the network are as follows:



If a Molecule of  $M_{sp}$  is added to the food, then an Interaction of each of the four types can take place in sequence (C1, C2, F3, F4). The overall scheme for this sequence of Interactions is:



The sequence can only proceed if at least one Molecule of  $M_{sp}$  is present, but once the reaction has started it continues due to the excess production of  $M_{sp}$ .

There is nothing ‘special’ about the sequence C1, C2, F3, F4. If the Interactions are considered in different sequences

<sup>6</sup>Molecule Type name convention:  $M_{sp}$ ,  $P_{sp}$  and  $D_{sp}$  indicate a monomer, polymer and ‘closed dimer’ respectively. See the section covering molecular structure for further explanation.

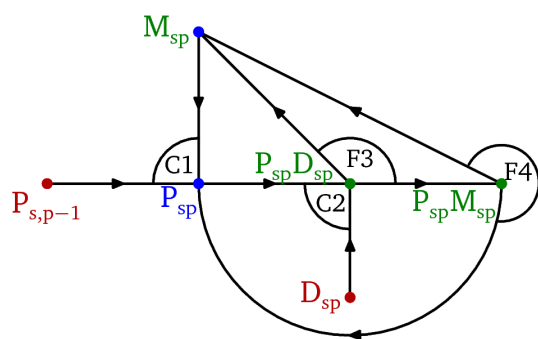


Figure 6: A memory unit for the ‘memory bank’. Molecule Types are colour coded as follows: red - input to this unit, blue - output from this unit to the next unit, green - intermediate products. The blue and green Molecule Types together form an autocatalytic set. If the inputs are present, a Molecule of any member of the set can activate the network.

then it can be seen that a single Molecule of any one of  $P_{sp}$ ,  $P_{sp}D_{sp}$  or  $P_{sp}M_{sp}$  (in addition to the food) is also sufficient to activate the network.

In short  $M_{sp}$ ,  $P_{sp}$ ,  $P_{sp}D_{sp}$  and  $P_{sp}M_{sp}$  are an autocatalytic set that can be activated by any member of the set.

## A Memory Bank Network

Figure 7 shows a ‘memory bank’ of 25 units in five independent rows or *series*. Each unit has a label  $U_{sp}$ , where  $s$  indicates the series, and  $p$  indicates the *position* of the unit in its series. Each unit has the structure shown in Figure 6, with only the specific Molecule Types varying. The large circles on the left of the diagram represent a maintained food set.

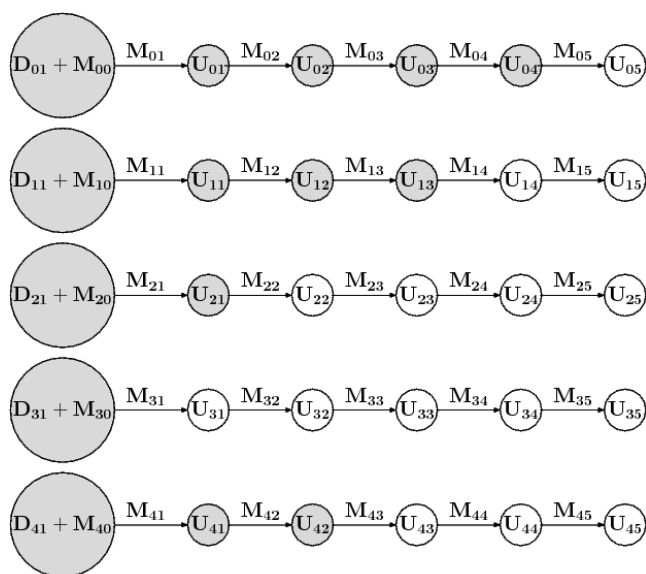


Figure 7: A Memory Bank with 25 units.

In each series the food provides the input to the first unit, and the outputs of each unit provide the inputs to the next unit. Each unit in a series may be either active or inactive; shading indicates an active unit. The next unit in a series can only become active if its predecessor is active (the maintained food set is considered to be the predecessor of the first unit, and is always active). The labels of the form  $M_{sp}$  over the arrows represent Molecule Types that will, if introduced in very small quantities, activate unit  $U_{sp}$  provided its predecessor is active.

Overall, the diagram represents a static network in which each of the five series has 6 possible states (from no units active, to all five active), so that the network as a whole can have  $6^5 = 7776$  different states. A network with ten series of nine units would have  $10^{10}$  possible states.

## Molecular Structure For The Memory Units

In this section the link between network structure and molecular structure is made. A set of SimSoup Molecule Types that produce the memory bank of the previous section is described.

## Molecular Structure in SimSoup

The approach to modelling molecular structure has been described elsewhere (Gordon-Smith, 2009b). It is summarised here, and an extension introduced for the work discussed here is described.

Molecules are two dimensional rigid structures built from Atoms bonded together such that they occupy fixed positions on a square ‘Board’ (similar to a chess board). Each square contains at most one Atom. Bond angles are always  $90^\circ$  or  $180^\circ$ , and bond lengths are all equal. Atoms bond together in a way broadly consistent with valence bond theory.

Molecules can Join or Split to form Molecules of different types. Joining must respect the ‘one Atom per square’ rule. Splitting occurs by breaking the weakest set of bonds that hold the Molecule into a single unit.

Bond strengths are usually fixed according to the types of Atom at each end of a bond. The extension introduced for this work is that in some cases, a bond can be perturbed (weakened or strengthened) by the proximity of Atoms that do not themselves participate in the bond.

## Atom Types For The Memory Unit Molecules

The SimSoup Atom Types used for the Memory Unit Molecule Types are described below:

- **Assemblite:** Forms two bonds. Can be used to assemble the structural framework for a Molecule. Colour: black
- **Stoppite:** Forms one bond, and when present at a bonding site stops further growth of the Molecule at that site (much as Hydrogen does in an organic molecule). Colour: grey

- **Junction:** Forms three bonds. Can be used to provide a 3 way junction in a structure. Colour: **blue**
- **Loosium<sup>7</sup>:** Forms three bonds. Can provide a weak (loose) bonding site within a structure. Does not bond to Anti-Loosium. Colour: **spring green**
- **Anti-Loosium:** Forms three bonds. Can provide a weak (loose) bonding site within a structure. Does not bond to Loosium. Colour: **cyan**
- **Grabite:** Forms three bonds. Can provide a bonding site in one monomer for another monomer to 'grab' as part of building a polymer. Colour: **red**
- **Hookite:** Forms two bonds. Can provide a 'hook' that can attach to an atom of Grabite to form a bond as part of building a polymer. Colour: **green**
- **Perturbium:** Forms three bonds. Bonds can be weakened or strengthened by nearby Metal atoms. Colour: **magenta**
- **Metal:** Forms one bond. Can perturb nearby Perturbium/Perturbium bonds, even though not bonded to Perturbium. Colour: **orange**.

## Memory Unit Molecule Structures And Dimer Splitting

**Monomers, Polymers And Closed Dimers** This section describes the structures of Molecule Types that appear as Reactants for Constructions C1 and C2 in Figure 6.

Molecule Types of the form  $M_{sp}$  are *monomers*, those of the form  $P_{sp}$  and  $P_{s,p-1}$  are (short) *polymers*, and those of the form  $D_{sp}$  are *closed dimers*.

Figure 8 shows examples. Figure 8a shows monomer  $M_{01}$  and its structural units. The positions of the two recesses labelled  $S = 0$  and  $P = 1$  vary as the series  $s$  and position  $p$  indices vary. The recesses are called the *series recess* and the *position recess* respectively.

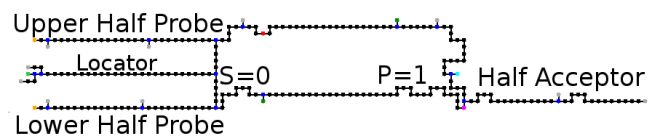
Along the top of each monomer are three small projections and a recess. The left hand *series projection* is directly above the series recess. The middle *position projection* is one place to the left of the position recess.

Figure 8b shows polymer  $P_{01}$ . The naming convention for polymers is such that  $P_{sp}$  represents a polymer of length  $p + 1$  whose end monomers are  $M_{s0}$  and  $M_{s,p}$ .

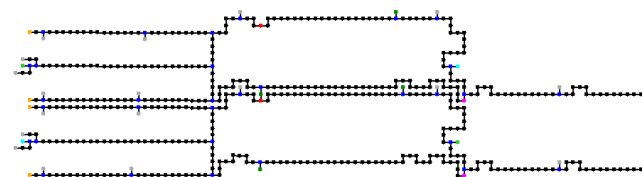
The positions of the recesses and projections on the top and bottom of the monomers ensure that two monomers can only join in a polymer if they are in the same series (same  $s$  index) and their position ( $p$ ) indices differ by 1.

The Half-Probes and Half-Acceptor on each monomer also have recesses/projections, and the positions of these are similarly dependent on the series and position indices.

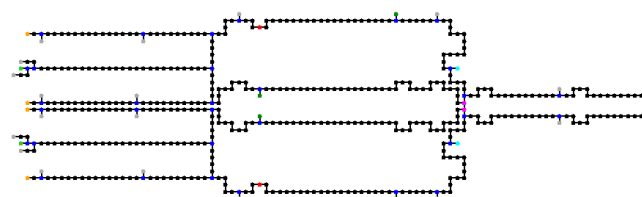
<sup>7</sup>Only two of the bonds supported by Loosium and Anti-Loosium are used for the memory unit Molecules.



(a) Monomer  $M_{01}$  showing structural units



(b) Polymer  $P_{01}$



(c) Closed Dimer  $D_{01}$



(d) Polymer  $P_{03}$

Figure 8: Example monomer, closed dimer, and polymers. See the supplementary material for larger examples.

The structure of the monomers allows for both  $s$  and  $p$  to vary between 0 and 9. There are therefore 100 possible monomer types, and these can be used to construct 10 series of polymers, with polymers in each series being built from up to 10 monomers. Each series corresponds to a row in an enlarged version of the memory bank of Figure 7.

Figure 8c shows a closed dimer, formed by joining two monomers 'back to back'.

Finally, Figure 8d shows  $P_{03}$ , a polymer of length 4.

**Dimer Splitting Intermediates And The Splitting Mechanism** Figure 9 shows the structure of the Fission Reactants in Figure 6. Figure 9a shows Molecule Type  $P_{01}D_{01}$ . Figure 9b shows Molecule Type  $P_{01}M_{01}$ .

Dimer splitting is a key mechanism for the memory unit. It provides the means by which the autocatalytic set of Figure 6 maintains itself. Taking the example of Figure 9,

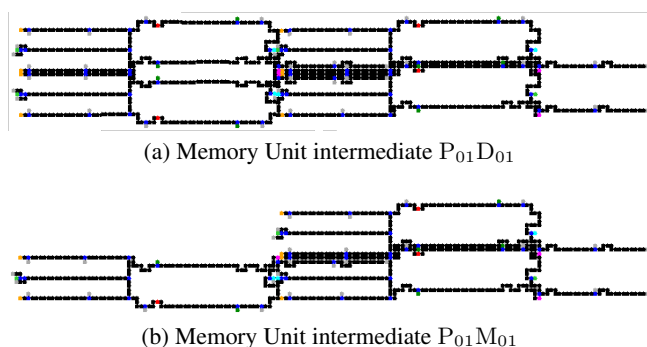


Figure 9: Dimer splitting intermediates

a Molecule of  $P_{01}D_{01}$  first splits (Fission F3) to release a Molecule of  $P_{01}M_{01}$  plus an  $M_{01}$  monomer, and then the  $P_{01}M_{01}$  Molecule splits (Fission F4) to release a  $P_{01}$  Molecule plus a second  $M_{01}$  monomer.

In short, the autocatalytic set maintains itself by splitting a ‘food’ dimer  $D_{sp}$  to produce a surplus of the monomer  $M_{sp}$ .

Dimer splitting involves a mechanism in which a polymer temporarily binds a dimer, and as result the Perturbium/Perturbium bond that holds the dimer together is weakened. The details of this can be explained by reference to Figure 10, which shows the central part of  $P_{01}D_{01}$ .

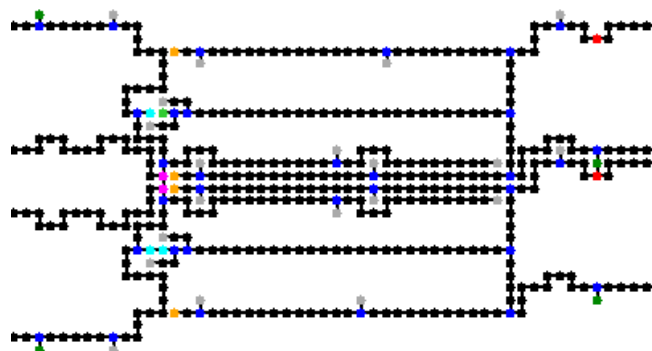


Figure 10: The central part of  $P_{01}D_{01}$ , showing the way in which the polymer ( $P_{01}$ ) part on the right ‘attacks’ and weakens the dimer ( $D_{01}$ ) part on the left at the bond between the two magenta Perturbium Atoms. The two parts of  $P_{01}D_{01}$  are held together temporarily by the weak bond between the two cyan Anti-Loosium Atoms.

The dimer and polymer parts are weakly bound at the Anti-Loosium/Anti-Loosium bond that joins the Locator of the  $M_{01}$  part of  $P_{01}$  to the bottom right of  $D_{01}$ . The memory mechanism relies on  $D_{sp}$  being split by  $P_{sp}$ , and not by any other polymer. ‘Incorrect’ splitting is ruled out because the two Half Probes on  $P_{sp}$  must be an exact match for the two Half-Acceptors on  $D_{01}$ .

The dimer weakening occurs because the two (orange)

Metal Atoms at the end of the two Half-Probes on the polymer are close to the two (magenta) Perturbium atoms on the dimer. This weakens the bond between them, and the dimer splits. The top ( $M_{01}$ ) part of the dimer falls away because it has no other bond either with the polymer or with the other ( $M_{01}$ ) part of the dimer. The other part of the dimer also spits from the polymer shortly afterwards, because the Anti-Loosium/Anti-Loosium bond holding the two together is weak, and so can only be temporary.

To summarise: A  $P_{sp}$  polymer binds temporarily to a  $D_{sp}$  dimer, and as a result the dimer is weakened. Both parts of the dimer separate from the polymer, which is then free to split another dimer. A dimer can only be split by the ‘correct’ polymer because the Probe and Acceptor formed by the Half-Probes and Half-Acceptors of the monomers involved must have compatible shapes.

## Results

Preliminary ‘proof of concept’ tests have been undertaken to investigate the workability of the memory bank described above. The tests used the SimSoup artificial chemistry simulator. Reactions take place in a well stirred Reactor. The rate constant  $k$  for Constructions is set to a constant value; those for Fissions are set to  $k = Ae^{-E_f/RT}$ , where  $E_f$  is the total energy of the bonds that have to be broken,  $T$  is temperature, and  $A$  and  $R$  are constants.

Results of two runs are presented. Both demonstrate memory; the first is typical of runs undertaken, the second illustrates an unusual ‘ringing’ phenomenon.

### Run 1

The scenario for Run 1 is as follows:

- Starting at time 1000, a constant supply of ‘food’ is provided to the Reactor. This consists of 400 Molecules of  $M_{00}$  every ten timesteps, plus 200 Molecules of each of  $D_{01}$ ,  $D_{02}$  and  $D_{03}$  every ten timesteps
- ‘Seed’ Molecules are added as follows: Five Molecules of  $M_{01}$  at time 10000, five Molecules of  $M_{02}$  at time 30000, five Molecules of  $M_{03}$  at time 50000
- At each timestep, every Molecule has a probability of 0.001 of being removed from the Reactor (‘leakage’)
- The size of Molecules was limited. This was necessary to enable the simulation to run within a reasonable time<sup>8</sup>

Figure 11 shows the numbers of the three polymers  $P_{01}$ ,  $P_{02}$  and  $P_{03}$  present in the Reactor at each timestep, along with the number of  $M_{00}$  Molecules<sup>9</sup>.

<sup>8</sup>The operation of SimSoup is such that whenever a new Molecule Type enters the Reactor as a result of two Molecules joining, all the possible ways the new molecular (‘board’) structure can interact with existing molecular structures must be calculated. This is computationally intensive.

<sup>9</sup> $M_{00}$  can be regarded as a polymer of length 1.

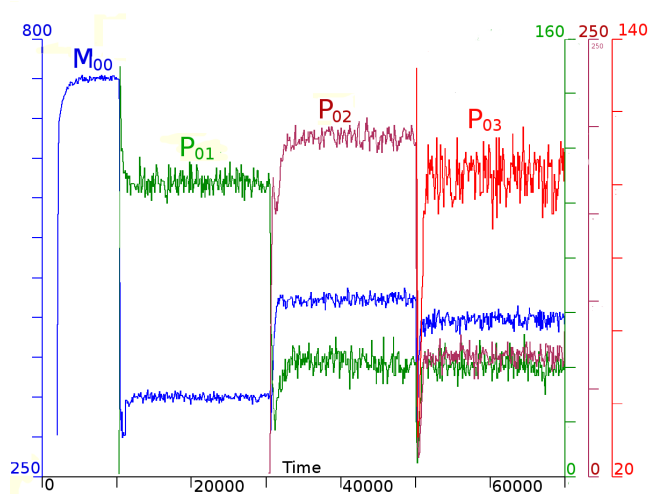


Figure 11: Plot showing the number of Molecules of  $M_{00}$ ,  $P_{01}$ ,  $P_{02}$  and  $P_{03}$  present in the Reactor during Run 1.

The addition of the ‘seed’ Molecules at times 10000, 30000 and 50000 in each case triggers a substantial change that persists over time. Prior to time 10000, there had been no Molecules of  $P_{01}$  present. Subsequent to the addition of the Molecules of  $M_{01}$  at that time, the number of  $P_{01}$  Molecules was stable at about 110 until time 30000.

Similar observations apply in regard to  $P_{02}$  and  $P_{03}$ . In each case, the seeding triggers a new persistent state in which the new Molecule Type is subsequently maintained.<sup>10</sup>

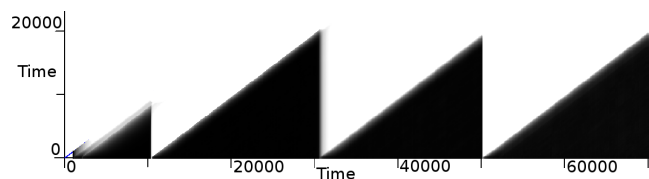


Figure 12: Manhattan Plot for Run 1. The black triangles indicate periods during which the Reactor composition (ie ‘mix’ of Molecule Types) varies little. The right hand edges of the triangles indicate sharp changes in composition.

Figure 12 is a ‘Manhattan Plot’ showing how the overall Reactor composition varied during the run. The construction of the Manhattan Plot has been described elsewhere (Gordon-Smith, 2007). The black triangles indicate periods during which there is little change in the composition (or ‘mix’) of Molecules in the Reactor.

The plot indicates that the pattern shown in Figure 11 in relation to a few key Molecule Types occurs more generally for the Reactor composition as a whole. There are periods

<sup>10</sup>The numbers of Molecules of existing types has a step change each time a new state is entered. This is to be expected since the overall dynamics of Interactions in the Reactor are changed. However, this does not lead to the disappearance of an existing type.

of roughly constant composition, and sharp changes corresponding to the addition of the ‘seed’ Molecules.

The number of Molecule Types present in the Reactor (not shown) was high; at the end of the run it was almost 500.

## Run 2

The scenario for Run 2 is similar to that for Run 1. There are differences in the timings at which Molecules are added.

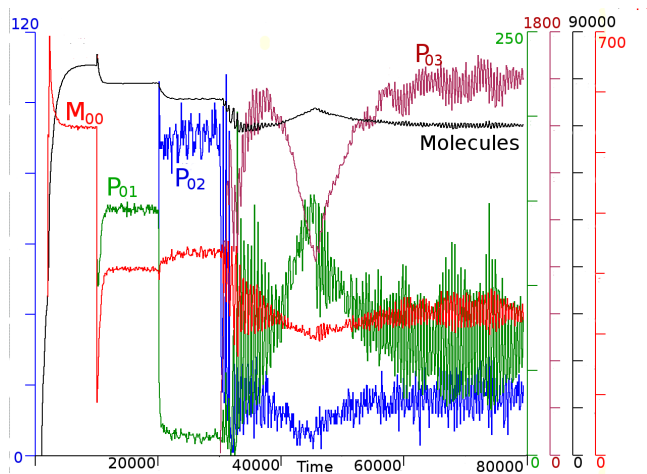


Figure 13: Time series plot for Run 2, showing ‘ringing’.

Figure 13 is a time series plot for Run 2. ‘Seed’ Molecules are added at times 10000, 20000 and 30000. The system ‘remembers’ each seeding as for Run 1 in Figure 11. However, after the third seeding the system shows a variable oscillatory or ‘ringing’ behaviour before stabilising.

## Discussion

**Stability:** The stability of the active state of a memory unit derives from the positive feedback mechanism that it incorporates. The design strategy for Molecule Types to support feedback is as follows. Firstly, identify each memory unit with a short polymer  $P_{sp}$  that can be produced from  $P_{s,p-1}$  by the addition of an  $M_{sp}$  monomer, and which can catalytically split a closed dimer  $D_{sp}$  to produce more  $M_{sp}$  monomers. Then design the monomers to join only in ways that lead to production of the ‘correct’ polymers, and ensure that these polymers only split the ‘correct’ closed dimers.

Transition of a memory unit from the inactive to the active state can be triggered by addition of just a single monomer. A suppression mechanism could be added if necessary for stability, although this would add model complexity.

**Moderate Complexity Of Monomers:** The designed monomers are moderately complex, although far below the complexity of DNA and RNA and the molecules involved in their replication. There may be scope for simplification. It can also be envisaged that they could be products of some

systematic process that would result in co-ordination of the positions of the various projections and recesses.

**Bias In Direction Of State Changes:** Changes in state of a Memory Bank in an organism only take place in the direction of increasing  $p$ . However, this does not mean that evolutionary ‘mistakes’ cannot be reversed. If an organism is less fit as a result of a mutation then it will be less likely to persist in future generations. It may be possible to change the design of the molecules to remove the bias, but it does not rule out open-ended evolution at the ecosystem level.

**Integration With Larger Network:** Although the Memory Bank consists of a number of independent rows (or series), it can be envisaged to be integrated within a larger metabolic network that it influences.

## Conclusions And Prospects

### Conclusions

- An artificial chemical network and associated molecular structures designed to support up to  $10^{10}$  persistent states has been shown<sup>11</sup>. It is reasonable to suppose that this would be sufficient for open-ended evolution to begin
- The monomers are of only moderate complexity, supporting the view that molecules with similar capabilities and properties (though no doubt very different structure) were present in the prebiotic world
- The operation of a small set of memory units has been simulated, completing the first part of the proof of concept
- Supplementary material for this paper is available at <http://www.simsoup.info/Publications.html>

### Prospects

- It will be appropriate to make optimisations enabling a larger set of Memory Units to be tested
- The author would like to hear from anyone interested in translating the ideas described here to ‘real’ chemistry.

## References

- Cairns-Smith, G. (1982). *Genetic takeover*. Cambridge University Press.
- Dittrich, P. and di Fenizio, P. S. (2007). Chemical organisation theory. *Bull. Math. Biol.*, 69(4):1199–1231.
- Dyson, F. (1999). *Origins Of Life*. Cambridge University Press.
- Fernando, C. and Rowe, J. (2007). Natural selection in chemical evolution. *J. Theor. Biol.*, 247:152–167.
- <sup>11</sup>The  $s$  and  $p$  indices both vary from 0 to 9. This gives 10 series each consisting of polymers of length between 1 (food) and 10.
- Gánti, T. (2003). *Principles of Life*. Oxford University Press.
- Gordon-Smith, C. (2005). SimSoup: An artificial chemistry model for investigation of the evolution of metabolic networks. ECAL Workshop. <http://www.simsoup.info/Publications.html>.
- Gordon-Smith, C. (2007). Evolution without smart molecules. ECAL Workshop. <http://www.simsoup.info/Publications.html>.
- Gordon-Smith, C. (2009a). The origin of life: A network oriented view. In *Proceedings - Levels of Selection and Individuality in Evolution: Conceptual Issues and the Role of Artificial Life Models, September 2009*.
- Gordon-Smith, C. (2009b). SimSoup: Artificial chemistry meets Pauling. In *Advances in Artificial Life: 10th European Conference, ECAL 2009, Proceedings*, Lecture Notes in Computer Science. Springer-Verlag.
- Jain, S. and Krishna, S. (1998). Autocatalytic sets and the growth of complexity in an evolutionary model. *arXiv:adap-org/9809003v1*.
- Kauffman, S. A. (1993). *The Origins Of Order*. Oxford University Press.
- Krishna, S. (2003). *Formation And Destruction of Autocatalytic Sets in an Evolving Network Model*. PhD thesis, Indian Institute Of Science.
- Oparin, A. (1957). *The Origin Of Life On The Earth*. Oliver And Boyd.
- Pauling, L. (1960). *The Nature Of The Chemical Bond*. Cornell University Press.
- Segré, D., Ben-Eli, D., Deamer, D. W., and Lancet, D. (2001a). The lipid world. *Origins Life Evol. B.*, 31:119–145.
- Segré, D., Lancet, D., Kedem, O., and Pilpel, Y. (1998). Graded autocatalysis replication domain (GARD). *Origins Life Evol. B.*, 28:501–514.
- Segré, D., Shenhav, B., Kafri, R., and Lancet, D. (2001b). The molecular roots of compositional inheritance. *Journal Of Theoretical Biology*, 213:481–491.
- SimSoup (2011). Simsoup artificial chemistry simulator source code. <http://code.google.com/p/simsoup/>.
- Wächtershäuser, G. (1990). Evolution of the first metabolic cycles. *Proc.Natl. Acad. Sci. USA*, 87:200–204.
- Wächtershäuser, G. (1997). The origin of life and its methodological challenge. *J. Theor. Biol.*, 187.
- Wächtershäuser, G. (2006). From volcanic origins of chemoautotrophic life to bacteria, archaea and eukarya. *Philos Trans R Soc London [Biol]*, 361:1787–1808.



# Clever Creatures: Case Studies of Evolved Digital Organisms

Laura M. Grabowski<sup>1</sup>, David M. Bryson<sup>2</sup>, Fred C. Dyer<sup>2</sup>, Robert T. Pennock<sup>2</sup> and Charles Ofria<sup>2</sup>

<sup>1</sup>University of Texas-Pan American, Edinburg, TX 78539

<sup>2</sup>BEACON Center for the Study of Evolution in Action  
Michigan State University, East Lansing, MI 48823  
grabowskilm@utpa.edu

## Abstract

We present a “bestiary” of three digital organisms (self-replicating computer programs) that evolved in three different experimental environments in the Avida platform. The ancestral environments required the evolving organisms to use memory in different ways as they gathered information from the environment and made behavioral decisions. Each organism exhibited a behavior or algorithm of particular interest: 1) simple step-counting odometer; 2) clever low-level computation; and 3) pronounced modularity in both program structure and program functionality. We present descriptive in-depth analysis of the case study organisms, with a focus on the structure and operation of the evolved algorithms that produce the individuals’ fitness enhancing behaviors.

## Introduction

The multi-disciplinary nature of Artificial Life (Alife) makes for rich cross-fertilization between computer science and biology, with research that focuses on organizing principles of living systems (Bedau, 2007). In the broad context of evolving building blocks of simple intelligent behavior, we present experiments that address the interaction of memory, environment, and learning in an evolutionary context. Here, we recount the key points of work reported in Grabowski et al. (2010) and expand on that discussion. In the previous paper, we presented our experimental motivation and design, and gave a high-level view of the evolved behavior. In the current paper, we dissect the algorithms that lay beneath the evolved behaviors, exposing the low-level mechanisms that produce the fitness enhancing behaviors. We produced our analyses through instruction-by-instruction examination of execution traces of the evolved digital organisms.

An important aim of our approach is to inform inquiry in both computer science and biology. With that aim in mind, we selected three highly successful digital organisms that evolved in three different experimental environments. Each of the case study organisms has a salient feature or behavior that seems critical for the evolved solution to work. The first organism evolved a simple odometer that it uses to count its steps, turning immediately before it would have

otherwise entered an environmental hazard. The second organism evolved a computational strategy that uses low-level bit operations to ensure correct behavioral responses to cues from the environment. This computational tactic is of special interest because it produces high-level behavior from low-level operations, and it also exemplifies a foundational principle of biology and psychology, that evolution tends to produce parsimonious solutions to behavioral problems (“Morgan’s Canon,” Morgan (1894)). The third organism evolved distinct functional and structural modularity; the role of modularity is a topic of great interest in a number of contexts. These digital organisms were products of an open-ended evolution system, and the system did not explicitly select for any of the solutions. We are exploring the range of unexpected solutions that can come out of such a system. The diversity of the evolved solutions is broad, even though we are dissecting only a handful of examples.

## Methods

### Avida: Overview

Digital evolution (Adami et al., 2000) is a type of evolutionary computation that places a population of self-replicating computer programs (digital organisms) in a computational environment, where the population evolves as the organisms replicate, mutate and compete for environmental resources. Digital evolution is a useful tool for understanding evolutionary processes in biology and for leveraging evolution to find solutions to computing and engineering problems. Avida (Lenski et al., 2003; Ofria et al., 2009) is a widely used software platform for digital evolution. Avida is an instance of evolution in its own right (Pennock, 2007), and provides a host of tools for experimental studies. In this section, we provide a brief summary of how Avida functions. For more detailed information, see Ofria et al. (2009).

The Avida world is a discrete two-dimensional grid of cells that holds the population of digital organisms. At most one organism (Avidian) may occupy a grid cell. The genome of an Avidian is a circular list of program instructions that resemble assembly language, that runs in a virtual central processing unit (CPU). The organism’s CPU contains three



registers (AX, BX, and CX), two stacks, and several heads (FLOW, used as a target for jumps; IP, an instruction pointer that denotes the next command to be executed; READ, for reading an instruction; WRITE, for writing an instruction). Execution of the instructions in the organism's genome act on the elements of the virtual CPU, incurring a cost measured in virtual machine cycles. An Avidian accomplishes all functions by executing the instructions in its genome, such as movement, gathering information from its environment, or replicating. The basic Avida instruction set is Turing-complete (Ofria et al., 2002), and is easily extended by adding new instructions to the system.

An Avidian replicates by copying its genome into a new block of memory. Mutations in Avida occur through errors in this copying process that produce differences between the genomes of parent and offspring. These differences may take the form of inserting or deleting an instruction, or changing one instruction to another, and occur at random with a user-defined probability. The Avida instruction set has the property of remaining syntactically correct in the presence of mutations, so a mutated genome will continue to execute, even if it performs no useful functions (Ofria et al., 2002).

Newly-produced offspring are placed in a randomly selected grid cell, overwriting any organism that was occupying the cell. This process gives a fitness enhancing advantage to an organism that can replicate faster than others in the population; organisms compete for the limited resource of grid space, and individuals that replicate sooner than others will have a higher proportion of descendants in future populations. Avidians may replicate sooner if they speed up their execution by accumulating metabolic rate bonuses as they evolve to perform user-specified tasks. Fitness in Avida is measured as the organism's metabolic rate divided by the number of cycles the organism requires to replicate.

## Experiment Design

We placed each Avidian in an environment containing a path that it could follow to collect food and increase its metabolic rate. Our environments were inspired by maze-learning experiments with honey bees (Zhang et al., 2000). Organisms had to sense the cues that formed the path and react appropriately to them. In some cases, advantageous behavior involved the ability to store experience for later decision-making.

For these experiments, we added sensing and movement instructions to the basic Avida instruction set. The *sg-move* instruction allows an organism to move one cell in the direction of its current orientation (its facing). In this study, each digital organism had its own virtual grid, so organisms did not interact during movement. Two instructions accomplished orientation changes, *sg-rotate-right* for turning 45° to the right and *sg-rotate-left*, for turning 45° to the left.

We added a sensing instruction, *sg-sense*, that allowed the

Avidian to get sensory information from its environment. When an Avidian executes the sensing instruction, the instruction places a predefined value in the executing Avidian's BX register, according to which cue is present in the grid cell at the organism's current location. These values are analogous to sensory input that the organism obtains from the environment, and are not directly used in calculations. The operation of this *sg-sense* instruction is important to the analyses of the evolved programs. The virtual grids for these experiments had a sensory cue in each cell of the grid. The environments contained some combination of the following cues (Grabowski et al., 2010):

1. **Nutrient:** A cue that indicates a cell is on the path, and provides "food" (*i.e.*, energy that adds to the organism's metabolic bonus). The nutrient cue has a sense value of 0 from the *sg-sense* instruction.
2. **Directional cue:** A cue indicating that a 45° turn to either the right or left is needed to remain on the path; the cell also contains nutrient. Right turns and left turns have different sense values from *sg-sense*, 2 for right and 4 for left.
3. **General turn cue:** A cue that indicates a turn but does not specify the direction, and contains nutrient. The return value for the general turn cue is 1.
4. **Empty:** A cue that indicates a cell that is not on the path. Movement into empty cells depletes energy gained by movement into cells that are on the path. The *sg-sense* instruction returns a sense value of -1 for empty cells.

We added two new comparison instructions to the Avida instruction set, *if-greater-than-X* (*if-grt-X*) and *if-equal-to-X* (*if-equ-X*), that supplemented existing comparison instructions. These instructions allow an organism to compare the value in its BX register to a predefined value. A *no-op* (NOP) label immediately following the comparison instruction determines the value to use in the comparison. We added the new comparison instructions because an Avida organism has to combine several different arithmetic instructions in order to compare a register value to any specific value. The new *if-equ-X* and *if-grt-X* instructions provided a shortcut and simplified comparisons for the Avidians, and also contributed to evolved genomes that were simpler to analyze. The details of these new instructions did not adversely affect the adequacy of our model, since our focus in the experiments was on memory; the mechanisms of constructing comparisons are not relevant to our questions of interest.

We constructed several environment types using the cues described above. The three organisms that we present in this paper evolved in three different environments.

- **Environment 1 and Environment 2: Evolving reflexes.** The first two environments contain paths with directional

(right and/or left), nutrient, and empty cues. Paths in Environment 1 contained only one type of directional cue (right or left) in each path instance; they are “single-direction turn” environments (see Figure 1). Environment 2 paths contained both right and left turns in the same path, and so are “dual-turn” environments (see Figure 2a). With both of these environment types, we expected the organisms to evolve a reflexive reaction to the path cues that they sensed, since the sensory information did not have to be retained for decision-making in the individual’s future.

- **Environment 3: Evolving volatile memory.** This environment type uses all four sensory cue types. The specific directional cue (right or left) is encountered when the turn is the first turn on the path or when the turn direction changes (*e.g.*, the organism has done one or more right turns and now needs to turn left). The general turn cue is encountered when the turn direction is to remain the same as the previous turn (*e.g.*, the organism executed a left turn at the previous turning and the current turn is also to the left) (see Figure 2b). This arrangement requires the organisms to evolve mechanisms for storing, using, and updating information about their experience on the path they are traversing, equating to a simple form of memory.

Organisms were presented with one of several different paths of the particular environment type (four different paths for Environments 1 and 3, and five different paths for Environment 2), chosen at random when the organism was born. Each individual experienced only one specific path in its lifetime, but all of the environments were experienced by multiple organisms during the course of evolution. In all experiments, organisms could raise their metabolic rate bonus through a path traversal task. The details of the task are presented in Grabowski et al. (2010). We ran 50 experimental replicates for each environment type, seeding each experiment with an organism with only the ability to replicate. All other functions had to evolve, using instructions entering the organism’s genome through mutations. We used the default Avida mutation rates for all our experiments, a 0.085 genomic mutation rate for a length-100 organism (a 0.0075 copy-mutation probability per copied instruction, and insertion and deletion mutation probabilities of 0.05 per divide) (Ofria et al., 2009). Experiments ran for a median of approximately 33,000 generations (250,000 Avida updates). Our populations had a maximum of 3600 individuals.

## Results and Discussion

### Environment 1: Evolved Odometry

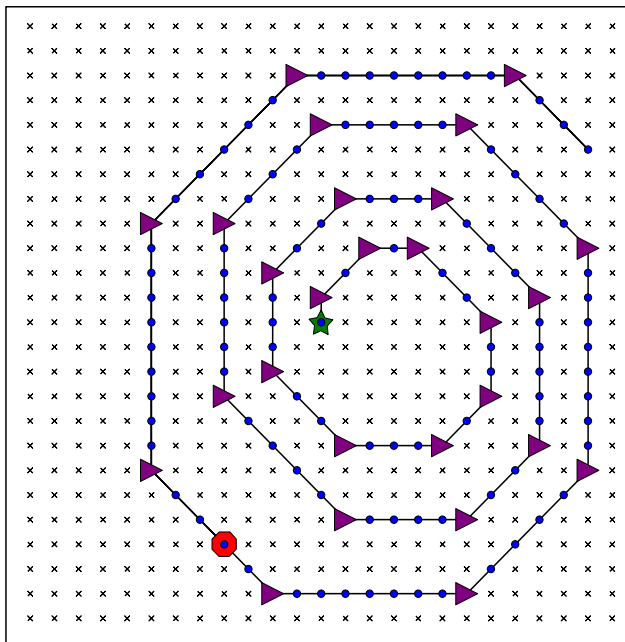
For Environment 1 (single-direction turn paths), we deliberately constructed simple paths with two regularities: each individual environment contained only right turns or only left turns, and the path progressed continuously outward from the starting position, giving the paths a spiral shape. There

was also one unintentional regularity: the ancestral right-turn paths were the same except for the organism’s starting position and the resulting distance to the first turn. The left-turn paths had more differences in the numbers of steps between turns. One population from this environment evolved a step-counting organism. This result is particularly exciting, since some animals use a mechanism analogous to step counting to determine the distance they have traveled on excursions away from their nests (Wittlinger et al., 2006). While odometry is considered a straightforward problem in robotics, it is by no means clear how it works in most animals, how it participates in higher-level processes such as path integration, and how it first evolved. Our approach may afford a way of exploring these problems.

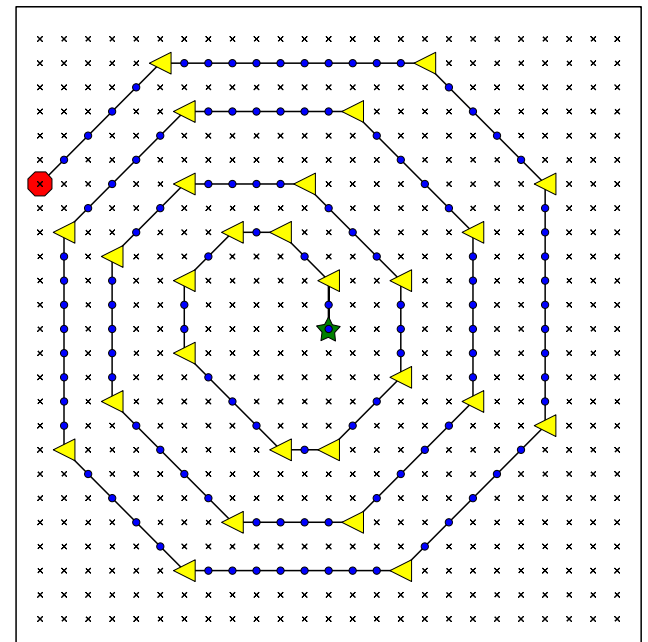
Figure 1 shows trajectories of the Environment 1 example organism (Org:StepCount) moving on a right-turn-only path (Figure 1a) and on a left-turn-only path (Figure 1b). Org:StepCount’s evolved strategy performed well in both turn environments. Interestingly, Org:StepCount backtracked on the right-turn grid, *i.e.*, it turned around and retraced its steps on the path. This behavior did not reduce Org:StepCount’s metabolic rate; the task quality calculation rewarded movement into unique path cells but did not penalize an organism for multiple movements into a path cell (Grabowski et al., 2010). Org:StepCount was able to navigate the entire right-turn path without entering any empty cells and also successfully followed the left-turn-only path, stopping after it encountered a single empty cell.

We analyzed an execution trace of Org:StepCount while it traversed each of these two paths, to uncover how its algorithm produces the observed behavior. Most—but not all—of the movement and replication code of Org:StepCount’s program is organized into two sections. Some instructions for this behavior (*i.e.*, movement and replication) are scattered in other locations in the genome, so Org:StepCount is not completely modular. A distinctly modular organism evolved in Environment 3, discussed later. One of the code sections (“Section 1A”) handles moving on a right-turn path, and the second (“Section 1B”) focuses on left-turn paths. Section 1B also contains a nested copy loop that is used for replication. Both of these code sections execute, whether the organism is on a right-turn or left-turn path, but the resulting behavior differs according to the path type (*i.e.*, right or left). Section 1A is essentially a counting routine. When Org:StepCount is traversing a right-turn path, Section 1A counts its steps; for left-turn paths, Section 1A counts the number of 45° turns the organism executes. When on a left-turn path, Org:StepCount uses Section 1B to travel to the end of the path and then replicate. When Org:StepCount is on a right-turn path, Section 1B allows the organism to avoid stepping off the end of the path by retracing some of its steps, at the same time finishing its replication process.

The following is a pseudocode description of the functionality of Section 1A:



(a) Right Turn Path



(b) Left Turn Path

Figure 1: Trajectories of Org:StepCount on ancestral paths.

```
DO
  rotate right
  IF (CX > 0) copy
  copy
  CX <- sense
  IF (CX equal nutrient) rotate left
  ELSE IF (CX equal right turn)
    CX <- 128
    move
    BX <- BX + 1
  WHILE (BX not equal CX)
```

Org:StepCount's current environment (*i.e.*, left- or right-turn) determines how this code executes. When traversing a right-turn path, Org:StepCount uses this loop to count its steps to the end of the path. Setting the CX register to the value of 128 (by reading the current position of the Instruction Pointer (IP)) and incrementing the value in the BX register (which begins at a value of 0 at the first loop iteration) with every loop iteration sets up the exit condition for the loop: after Org:StepCount has taken 127 steps in the loop, the last increment of the BX register causes execution to exit the loop. When executing this loop on a left-turn path, the organism remains in the same spot and executes the loop four times, performing a one-eighth turn in each iteration. When the value of the BX counter reaches 4, Org:StepCount exits the loop, and is now facing in the “wrong” direction (*i.e.*, facing back the way it has already come). The section

of code immediately following this section includes another set of four one-eighth turns, so Org:StepCount regains the facing it had upon entering Section 1A.

Section 1B operates as follows:

```
DO
  move
  BX <- sense
  IF (BX not equal nutrient)
    rotate left
  IF (BX equal empty)
    WHILE (not end label) copy
  ELSE IF (not end label) copy
  IF (end label) divide
  WHILE (BX not equal empty) AND
    (not end label)
```

When this algorithm is executed on a left-turn-only path, Org:StepCount moves along the path, eventually moving one step off the end of the path into an empty cell. At that point, Org:StepCount “stands still,” and executes a tight copy loop to complete copying its genome to its offspring, at which time it divides. On a right-turn path, however, Org:StepCount never enters the tight copy loop; instead, it copies just one instruction for each iteration of Section 1B, while it retraces its steps along the path. This strategy produces the backtracking in the trajectory plot of Figure 1a. The organism retraces the path moving back toward its initial location, stopping part of the way through the path (red

octagonal symbol). The number of instructions needed to produce an offspring remain similar on right- and left- turn paths (1779 instructions for the right-turn path shown in Figure 1a and 1780 instructions for the left-turn path shown in Figure 1b) since an extra instruction is copied with every iteration of Section 1A when Org:StepCount is moving on a right-turn path. Table 1 lists the Avida instructions for the two code sections described above.

Section 1A	Section 1B
sg-rotate-r	sg-move
if-grt-0	sg-sense
nop-C	nop-B
h-copy	if-n-equ
h-copy	sg-rotate-l
sg-sense	if-equ-X
nop-C	pop
jmp-head	if-less
sg-rotate-l	h-search
if-equ-X	if-label
get-head	nop-C
sg-move	h-divide
inc	h-copy
if-n-equ	mov-head
mov-head	

Table 1: Avida instructions for Org:StepCount.

## Environment 2: Economical Code and Clever Math

Environment 2, the dual-turn environment, presents evolution with a slightly more complex version of the problem encountered in Environment 1, since evolution must always contend with both turn directions in every path. The evolved algorithm of the example organism from this environment (Org:BitOperator) is interesting because it evolved some remarkably clever math that helped it succeed in its environment, using simple, low-level computations to produce complex, high-level behavior.

Org:BitOperator successfully negotiated both ancestral paths and novel paths. Figure 2a shows Org:BitOperator's trajectory on a novel path. The dimensions of the grid containing the path are different from the dimensions of the grids in the ancestral environments: the novel path shown has dimensions of  $20 \times 20$ , as opposed to the  $25 \times 25$  grids that were experienced during evolution. Since all environment grids are toroidal, the grid dimensions should make no difference to organisms, and organisms never have access to any global information. However, we included tests like these to provide additional evidence that the evolved algorithms do not work by finding and exploiting geometrical information, such as grid size, but instead function through gathering and using information from the environment.

Org:BitOperator executes most of its movement with a

concentrated movement loop. At a high level, the structure of the code is *move-sense-decide*. The decision concerns whether or not to turn, and if a turn is to be made, which direction to turn. Within the loop, conditional statements guard the turn directions to provide the correct execution flow for each environmental cue. In pseudocode, this movement loop functions as follows:

```
DO
  IF (BX > 1) rotate left
  copy
  move
  BX <- sense
  BX <- right-shift(BX)           #Line 1
  IF (BX equal 1) rotate right #Line 2
  ELSE IF (BX < CX)               #Line 3
    IF (BX > 0) CONTINUE          #Line 4
  WHILE (BX > 0)
```

Org:BitOperator has a simple, but clever, mechanism for using the default behavior of the comparison instructions to select the correct action, based on the current sense information. Org:BitOperator manipulates the current sensed cue value so that the values match the comparisons as needed. The key detail of this loop's execution is how the right-shift operation (Line 1) prepares the sensed cue value for use with the unmodified comparison statements. Stepping through the algorithm, starting from the `BX<-sense` line, the current cell cue is sensed, and the value placed in BX. That value is then right-shifted, dividing most sense values by 2. Recall the return values from the *sg-sense* instruction. If the sensed cue is nutrient (return value = 0), BX is still 0; if the cue is right-turn (return value = 2), BX is now 1; if the cue is left-turn (return value = 4), BX is now 2; if the cue is empty (return = -1), BX is still -1 (since the operation is an arithmetic right-shift, the sign bit is preserved in the shift). This low-level manipulation of the sense value permits the algorithm to use the default behavior of the comparison instructions, thus avoiding the need for NOP modification of the instructions. This characteristic provides more robust performance for Org:BitOperator, since the comparison needs only one instruction to complete its action, not two. The first comparison (Line 2) is true when the last sensed cue is right-turn, so the right turn is executed. The next comparison (Line 3) is false for all cues except empty, so execution returns to the top of the loop as long as the organism encounters non-empty cells. Sensing an empty cell triggers loop exit.

This solution is simple and economical, accomplishing the job with few extraneous instructions. Org:BitOperator has evolved an equally frugal copy loop near the end of its genome. The copy loop performs the bulk of Org:BitOperator's replication, and begins execution only after the movement loop has terminated. Table 2 gives the Avida code for Org:BitOperator's movement loop. Not only is the elegance of these evolved solutions to be admired from

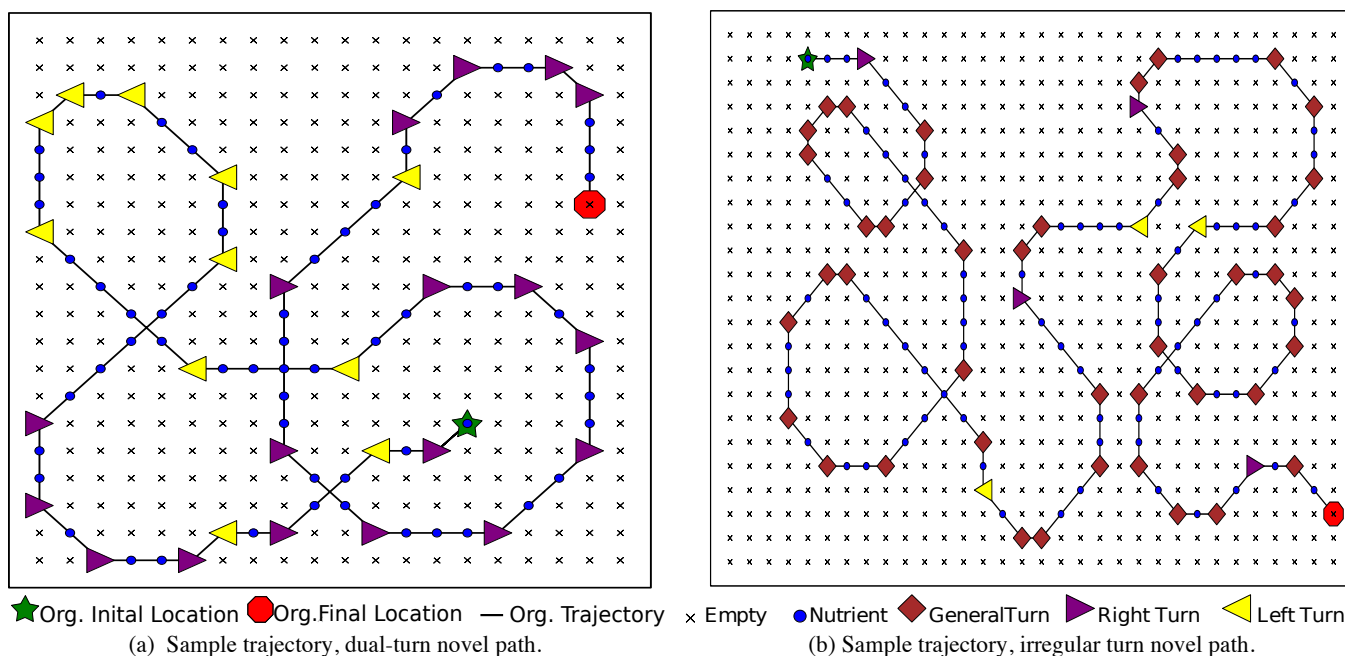


Figure 2: Trajectory of the two example evolved organisms from Environments 2 and 3. Figure 2a shows the example organism, Org:BitOperator, from Environment 2 (dual-turn paths) traveling on a novel path. The grid containing the path has different dimensions ( $20 \times 20$ ) from those of the ancestral paths ( $25 \times 25$ ). Figure 2b shows the example organism, Org:Modular, from Environment 3 (irregular paths) traversing a novel path. This path grid also has dimensions ( $23 \times 32$ ) that differ from those of the ancestral environments ( $25 \times 25$ ).

Movement Loop
if-grt-X
sg-rotate-l
h-copy
sg-move
sg-sense
shift-r
if-equ-X
sg-rotate-r
if-less
if-grt-0
mov-head

Table 2: Avida instructions fused in our example dual-turn environment organism, Org:BitOperator.

a computational perspective, they also provide evidence of “Morgan’s Canon,” a parsimony principle that has guided a century of research in animal and human psychology (Morgan, 1894). By this principle, one should prefer hypotheses that invoke simpler rather than more complex mechanisms of information processing. Our results suggest that digital evolution could lead to empirical study of this principle.

### Environment 3: Evolving Modularity

Environment 3 was the most complex environment in our study. To enhance fitness in this environment, organisms needed to make decisions based on their life experience, and update their memory of that experience at irregular intervals. The case study organism from this environment (Org:Modular, shown in Figure 2b traversing a novel path) evolved an algorithm with functional and structural modularity that provides appropriate behavioral responses to environmental conditions.

The execution of Org:Modular’s genome is fairly complex, with a high degree of flexibility to handle conditions in its environment. In general, Org:Modular moves its execution to different parts of its genome depending on the sensed cue from the environment. Org:Modular has two loops for its path-following, one loop that navigates left-turn path segments, “Module 3A,” and the other loop for traveling on right-turn path sections, “Module 3B.” Org:Modular has well-defined functional and structural modularity in its genome for handling right-turn and left-turn path sections. Such refined modularity was not observed in other organisms that we analyzed. Module 3A appears first in Org:Modular’s genome, before Module 3B. Module 3A can perform an arbitrary number of forward steps and consecutive left turns. This behavior in Module 3A is produced by a nested loop that results in straight-ahead movement on

the path; iterations of this smaller loop continue until a non-zero cue is sensed. The smaller loop terminates when a left-turn or general turn is sensed, but execution remains within Module 3A. Sensing a right-turn or empty cue will exit both the smaller loop and Module 3A. Module 3B enables Org:Modular to negotiate right-turn path sections, accommodating any number of repeated right turns and forward steps. Execution exits Module 3B upon sensing a left turn cue, and jumps to the beginning of the organism's genome, thereby arriving again at Module 3A. Execution of Module 3B terminates if an empty cell is sensed, continuing with the instructions following the module. Org:Modular also has a modular copy loop near the end of its genome that manages the majority of the copying for the organism's replication.

Module 3A, for navigating on left-turn path sections, functions as follows:

```
DO
  DO
    move
    BX <- sense
    IF (BX < CX)
      swap (BX, CX)
    BX <- BX - 1      # Line 1
    WHILE (BX < CX)   # Line 2
      rotate left
  WHILE (BX equal turn) OR (BX < CX)
```

The decrement of the value in BX following the sense instruction (Line 1) manipulates the value in the BX register such that execution remains in the nested loop as long as the organism is sensing nutrient cues (meaning that Org:Modular is moving straight on the path), but will exit the nested loop when any other cue is sensed. Whenever this module is executing, the value in CX is 0 at the top of the loop. Executing `BX <- BX-1` with the nutrient return value (0) places a value of -1 in BX, so execution does not exit the nested loop (Line 2). Decrementing the general turn cue return value (1) places a value of 0 in BX, causing execution to exit the nested loop and do the left turn. When the right-turn return value (2) is decremented, the value in BX becomes 1, and the nested loop is exited. Execution then exits Module 3A, after executing the left turn. The swap of values in BX and CX is executed only if an empty cell is sensed. The swap puts 0 in BX, and -1 in CX, so the nested loop is exited, and execution leaves Module 3A after the left turn, since BX is equal to CX after BX is decremented.

A pseudocode description of the functionality of Module 3B, for moving through right-turn path segments, is:

```
DO
  rotate right
  IF (BX < CX) BX <- sense
  move
  BX <- sense
  IF (BX equal turn) CONTINUE # Line 1
```

```
ELSE IF (BX equal left turn)
  jump IP to 0
  BX <- BX + 1
  rotate left
WHILE (BX not equal CX)
```

There is a section of instructions between the modules that has no move instructions, but has a single right-turn instruction that negates the last left turn performed before exiting Module 3A. An additional right-turn instruction executes before Module 3B entry, ensuring proper orientation for turning right, since Module 3B contains both right- and left-turn instructions that always execute. Correct orientation is maintained by selectively executing the left turn at the end of the module. When a general turn cue is sensed, execution in Module 3B skips the left turn (since `BX=1`), and returns directly to the top of the loop. When a nutrient is sensed, `BX=0`, so the increment of BX and the left turn are executed. When Org:Modular senses a left-turn cue, execution jumps out of Module 3B, returning to the beginning of the genome. As in Module 3A, the value in CX is 0 during execution of Module 3B. If an empty cell is sensed, incrementing the value places a value of 0 in BX, and execution exits the module. Once Org:Modular moves into an empty cell, execution moves to the copy loop, and Org:Modular completes its replication. Table 3 lists the Avida code for Org:Modular's path-following modules.

Two features of Org:Modular are particularly interesting. The first is the organization of the genome. The sections of the genome that do the bulk of the relevant behavior for Org:Modular—the two movement modules and the copy module—are functionally and spatially modular. For all three of these modules, very little happens within them apart from the main function of the module. The modules are also spatially modular, *i.e.*, located in different areas of the genome. Example organisms from the preceding experiments also demonstrate some structural modularity, but their functional modularity is less well-defined. The parallel with the structural and functional modularity seen in the neural control of animal behavior is striking (Bullmore and Sporns, 2009). The second feature of special interest is the flexibility of execution flow between code modules. The execution flow enables Org:Modular to cleverly handle all the contingencies of the environment. For example, even though Module 3A (left-turn module) is encountered first in the sequential execution of the genome, if a right turn is encountered first, the execution flow moves easily through Module 3A into Module 3B (right-turn module). The algorithm evolved to deftly maneuver along the paths, using the information of the cues from the environment to alter its execution.

We presented a “bestiary” of digital organisms, case studies of three evolved organisms that show the range of surprising solutions that can arise in open-ended evolving systems. Each example organism had a striking characteristic that highlights issues of interest in both computer sci-

Module 3A	Module 3B
sg-move	sg-rotate-r
sg-sense	if-label
sub	nop-B
if-less	add
swap	nop-B
h-divide	if-less
dec	sg-sense
if-less	sub
mov-head	sg-move
push	nand
if-label	sg-sense
nop-C	if-equ-X
shift-r	mov-head
nop-A	
sg-rotate-l	
if-equ-X	
if-less	
mov-head	

Table 3: Avida instructions for example irregular path organism, Org:Modular.

ence and biology. Although it is premature to make broad generalizations based on our results, we can conclude that the Avidians evolved solutions that were well tailored to the task, neither more nor less complex than needed. The lesson of our results is that an evolutionary approach to higher levels of intelligence will require careful attention to both the computational resources available to the evolving system and to the complexity of the tasks presented by the environment.

The work that we report in this paper laid the foundation for several ongoing research projects. We are continuing our study of evolving navigation, including simple landmark navigation and vector navigation. The experiments discussed in this paper provide an excellent arena for investigating issues relating to historical contingency in evolution: how accidental changes to the genetics of a population shape the path of future evolution. Steps in evolution are thus dependent on prior history (Blount et al., 2008). We are exploring what factors determined what strategy evolved in our experiments.

Our results underscore how results from Artificial Life experiments can provide insight for different fields of study. The strategies shown by these case study organisms highlight the power of evolution to find surprising and clever solutions to problems. These solutions may guide the development of intelligent artificial agents, and also provide insights into the fundamental principles governing the early evolution of intelligent behavior in biological systems.

## Acknowledgments

We thank Philip McKinley, Wesley Elsberry, Jeff Clune, Michael Vo, Erica Rettig, and other members of the MSU Digital Evolution Laboratory for valuable insights and feedback. This work was supported by a grant from the Cambridge Templeton Consortium, “Emerging Intelligence: Contingency, Convergence and Constraints in the Evolution of Intelligent Behavior,” a grant from the DARPA FunBio program, and NSF grants CCF-0643952 and DBI-0939454.

## References

- Adami, C., Ofria, C. A., and Collier, T. C. (2000). Evolution of biological complexity. *Proceedings of the National Academy of Science*, 97:4463–4468.
- Bedau, M. A. (2007). Artificial life. In Matthen, M. and Stephens, C., editors, *Handbook of the Philosophy of Science: Philosophy of Biology*, pages 595–613. Elsevier, Boston.
- Blount, Z. D., Borland, C. Z., and Lenski, R. E. (2008). Historical contingency and the evolution of a key innovation in an experimental population of *Escherichia coli*. *Proceedings of the National Academy of Science*, 105(23):7899–7906.
- Bullmore, E. and Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198.
- Grabowski, L. M., Bryson, D. M., Dyer, F. C., Ofria, C., and Pennock, R. T. (2010). Early evolution of memory usage in digital organisms. In *Artificial Life XII: Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*, pages 224–231, Cambridge, MA. MIT Press.
- Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423:139–144.
- Morgan, C. L. (1894). *An introduction to comparative psychology*. Scott, London.
- Ofria, C., Adami, C., and Collier, T. C. (2002). Design of evolvable computer languages. *IEEE Transactions in Evolutionary Computation*, 17:528–532.
- Ofria, C., Bryson, D. M., and Wilke, C. O. (2009). Artificial life models in software. In Adamatzky, A. and Komosinski, M., editors, *Advances in Artificial Life*, chapter Avida: A Software Platform for Research in Computational Evolutionary Biology, pages 3–36. Springer-Verlag, Berlin, 2nd edition.
- Pennock, R. T. (2007). Models, simulations, instantiations, and evidence: the case of digital evolution. *Journal of Experimental and Theoretical Artificial Intelligence*, 19(1):29–42.
- Wittlinger, M., Wehner, R., and Wolf, H. (2006). The ant odometer: Stepping on stilts and stumps. *Science*, 312(5782):1965–1967.
- Zhang, S. W., Mizutani, A., and Srinivasan, M. V. (2000). Maze navigation by honeybees: learning path regularity. *Learning and Memory*, 7:363–374.



# EcoSim: an individual-based platform for studying evolution

Gras R.<sup>1</sup>, Golestani A.<sup>1</sup>, Hosseini M.<sup>1</sup>, Khater M.<sup>1</sup>, Majdabadi Farahani Y.<sup>1</sup>, Mashayekhi M.<sup>1</sup>, Sina M.<sup>1</sup>, Sajadi A.<sup>1</sup>, Salehi E.<sup>1</sup> and Scott R.<sup>1</sup>

<sup>1</sup> School of Computer Science, University of Windsor, ON, Canada  
rgras@uwindsor.ca

## Abstract

To study the evolutionary process and the emergence of species, we have conceived an individual-based evolving predator-prey ecosystem simulation presented in (Gras, 2009). One major and unique contribution of this simulation is that it combines a behavioral, an evolutionary and a speciation mechanism. This is the only simulation modeling the fact that individual behaviors affect evolution and speciation. We have already obtained some very interesting and promising results from our simulation on species abundance distribution, study of chaotic patterns or population spatial distribution.

## Introduction

Since the last decade, the individual-based modeling approach became more common as machine capable of running time-consuming simulations appeared (DeAngelis, 2005). However, few attempts have been made to simulate a complete and complex ecosystem. The first one is Echo (Hraber, 1997), which includes an evolutionary mechanism. However, the organisms are very simple, and have no behavior model. Another system studying long term evolution is Avida (Lenski, 1999). It has nevertheless limitations such as: the individuals do not move, are quite limited in number, and there is a fix fitness function which means that the system is mostly an optimization process.

Other models, such as PolyWorld (Yaeger, 1992), Bubbleworld.Evo (schmickl, 2006) or Framsticks (Comosinski, 2000), have been proposed including more complex agents and behavioral models. They use Artificial Neural Networks or system of learned rules to evolve the agent's behavioral model during their life and by an evolutionary process. However, these approaches are highly computational expensive and only allows small population (few hundred) of agents. They are therefore more dedicated to investigate evolution of learning capacities than high scale mechanisms involving populations and species dynamics. To investigate several open theoretic ecological questions we have designed, Ecosim<sup>1</sup> (Gras, 2009), a large scale simulation platform. Our general purpose is to study how individuals and local events can affect high level mechanisms such as community formation, speciation or evolution.

<sup>1</sup> <http://sites.google.com/site/ecosimgroup/research/ecosystem-simulation>

## Our model

To observe phenomena at the evolutionary scale that affect the individual behaviors, several constraints need to be fulfill: (1) every individual should possess genomic information that will be the subject of the evolutionary process; (2) this genetic material should affect the individual behavior and consequently its fitness; (3) it has also to be transmitted and modified from generation to generation; (4) a sufficiently high number of individuals should coexist and their behavioral model should be sophisticated enough in order that complex interactions and organizations could emerge; (5) a model for species representation and an speciation mechanism, leaning on the genomic and behavioral model, has to be defined; (6) for speciation events to occur and new co-adapted behavioral models to emerge and in turn affect the whole system, a large number of time steps need to be performed. We therefore face a computational challenge for both memory management and computational power. We need a model which allies the compactness and easiness of computation with a high potential of complex representation.

We have used a modified version of the Fuzzy Cognitive Map (FCM) model (Kosko, 1986) and adapted it to our problem. This model is used at the same time as the behavioral model of our agents (our individuals) and as the vector of transmission of the evolutionary information. It allows a combination of compactness with a very low computational requirement while having the capacity to represent complex high level notions. Therefore, each agent can possess its unique proper FCM, which is an inherited modified combination of the ones of its two parents. The system can still manage several hundreds of thousands of agents simultaneously with reasonable computational requirements.

The FCM contains sensitive concepts such as: predatorClose, foodClose, mateClose, energyLow; internal concepts such as: fear, hunger, sexualNeed, curiosity, satisfaction; and motor concepts such as: escape, searchForFood, socialize, eat, breed. It includes also weighted links representing the mutual influences of these concepts. Our simulation implements a speciation mechanism related to the genotypic cluster definition (Mallet, 1995). A species is a set of individuals associated with the average of the genetic characteristics of its members. A species split if the difference between the FCMs

of the two most dissimilar agents is greater than a threshold (Aspinall, 2010) leading to two sister species that rapidly diverge to become genetically isolated with no more interbreeding events. Since species membership is evaluated at each time step, a species can emerge or extinct at any time. A typical run last several tens of thousands of time steps, each time step being the time needed for each agent to perceive its environment, to use its behavioral model to make a decision, perform its action, to update the species and the world parameters. In total, more than one billion of agents will be born and several thousands of species will be generated, which allows the evolutionary process to take place and new behaviors to emerge reacting to a constantly changing environment. In addition, a food chain consisting of three levels, primary producers, predators and preys, has been implemented allowing complex interactions between agents and co-evolution to occur.

## Results

### Species abundance distribution

To validate EcoSim, we have compared the ecological patterns it generates with those observed in natural ecosystems (Devaurs, 2010). We have focused on species abundance patterns as they are a key component of ecological theories. To analyze them, we used Fisher's logseries, since it is one of the most classical models of species abundance distribution. The following results, well established in the ecological literature, are also observed in the communities generated by our simulation: the logseries presents a good fit to the distributions of small samples; it fails to do so for large samples and complete community; the logseries performs better on species-rich communities. Even though the logseries does not provide a good fit for large samples, the distribution patterns observed in our communities are very similar to those observed in nature. Thus, at any level, our simulation gives coherent results in terms of relative species abundance.

### Chaotic behavior

Any attempt to model a real system needs to have the capacity to generate patterns as complex as the ones of the real system. We have studied the properties of the time series representing the variation of the number of: preys, prey species, predators and predator species (Golestani, 2010). We examined whether a chaotic behavior exists in these signals. To enforce our results, we use four different methods: Higuchi fractal dimension, correlation dimension, largest Lyapunov exponent, P&H method. To obtain a statistically significant evaluation, we apply the surrogate test method on 24 samplings of these data. All of them providing clear predictions that the behavior of simulation is deterministic chaotic.

### Population spatial distribution

We have conceived a measure to compute the spatial center of a population in a torus world (Sina, 2011). Computing spatial distribution in an ecosystem simulation is important for analyzing various aspects of species or group of individuals.

One of the applications is prediction of extinction of a population. When individuals of a population are dying, their spatial distribution either globally or locally, starts to decrease; since it has a relationship with number of living individuals of the population. Also it was shown, as expected in the parapatric model of speciation that genetically similar individuals in a population tend to live closer to each other.

## Conclusion

This project is at its early stage but we have already many interesting results. We have submitted several papers: diffusion and mitigations of diseases in an ecosystem, a machine learning approach for modeling species abundance distribution, the natural selection effect on the variation of the ecosystem's complexity, and the multifractal properties of the individuals' spatial distribution. We are currently working on: the effect of reduction of gene flow on the rate of speciation, the emergence of new complex behaviors and their effect on fitness, applying machine learning techniques to predict species extinction and speciation events, the effect of choice of mating partner on variation of the population's fitness and the effect of multiple food resources on emergence of species.

## References

- Aspinall, A. and Gras, R. (2010). K-Means Clustering as a Speciation Method within an Individual-Based Evolving Predator-Prey Ecosystem Simulation, *Lecture Notes in Computer Science*, 6335:318-329.
- DeAngelis, D. L. and Mooij, W. M. (2005). Individual-based modeling of ecological and evolutionary processes, *Annu. Rev. Ecol. Evol. Syst.* 36:147.
- Devaurs, D., Gras, R. (2010). Studying species abundance distributions in an individual-based simulated ecosystem through Fishers log-series, *Simulation Modelling Practice and Theory*, 18:100-123.
- Golestani, A., Gras, R. (2010). Regularity Analysis of an individual-based Ecosystem Simulation, *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20 (043120).
- Gras, R., Devaurs, D., Wozniak, A. and Aspinall, A. (2009). An Individual-based Evolving Predator-Prey Ecosystem Simulation using Fuzzy Cognitive Map as Behavior Model, *Journal of Artificial Life*, 15(4):423-463, 2009.
- Hraber, P. T., Jones, T. and Forrest, S. (1997). The Ecology of Echo. *Artificial Life*, 3:165-190.
- Komosinski, M., Rotaru-Varga, A. (2000). From directed to open-ended evolution in a complex simulation model. In *Artificial Life VII*, pages 293–299. MIT Press.
- Kosko, B. (1986). Fuzzy cognitive maps, *International Journal of Man-Machine Studies*, 24(1):65–75.
- Lenski, R. E., Ofria, C., Collier, T. C. and Adami, C. (1999). Genome complexity, robustness, and genetic interactions in digital organisms, *Nature*, 400:661-664.
- Mallet, J. (1995). A species definition for the modern synthesis, *Trends Ecol. Evol.*, 10:294.
- Sina, M., Gras, R. (2011). Computation of population spatial distribution in individual-based ecosystem simulation, To appear in *IEEE ALIFE 2011*.
- Schmickl, T. and Crailsheim, K. (2006) Bubbleworld.Evo: Artificial Evolution of Behavioral Decisions in a Simulated Predator-Prey Ecosystem, *Lecture notes in Computer Science*, 4095:594-605.
- Yaeger, L. (1992). Computational Genetics, Physiology, Metabolism, Neural Systems, Learning, Vision, and Behavior or PolyWorld: Life in *Artificial Life III*, pages 263-298. MIT Press, Cambridge, MA.

# Promoting Meme Diversity and Transmission Fidelity in Artificial Proto-Cultures

Andrew Guest<sup>1</sup>, Andrew Sapeluk<sup>1\*</sup>, Alan Winfield<sup>2</sup> and James Bown<sup>1</sup>

<sup>1</sup> School of Computing and Engineered Systems, University of Abertay Dundee, UK

<sup>2</sup> Bristol Robotics Laboratory, University of the West of England, Bristol, UK

[\\*A.Sapeluk@abertay.ac.uk](mailto:A.Sapeluk@abertay.ac.uk)

## Abstract

Cultural evolution occurs through the transmission of cultural traits, and we consider the meme as the unit of cultural transmission. We construct an agent-based model representing the processes by which cultural transmission occurs and to link these to the community-scale phenomena arising from agent interactions. We base our model on small communities of e-Puck robots, and following work on movement-based memes, consider sound as the medium of cultural transmission. Our architecture affords (re)production of memes, variation in meme production and a range of meme selection and meme memory strategies. Through these processes, we identify the meme complexities, meme memory strategies, meme selection strategies and e-Puck movement speeds that promote and inhibit both meme diversity and reproductive fidelity.

## 1 Introduction

Human and animal decision-making is strongly influenced by knowledge acquired through observation of the behaviour of others, and when behavioural patterns are spread among individuals over generations this is a form of cultural evolution (Danchin et al., 2004). Thus cultural evolution occurs through the transmission of cultural behaviours / traits (Christensen and Kirby, 2003). Agent-based models seek to understand the processes by which cultural transmission occurs and to link these to the community-scale phenomena observed when groups of agents interact (Buzing et al., 2005). By understanding these processes at the individual scale it is possible to manage change at the community scale (Bown et al., 2007). To effect cultural change is a challenging problem and it is possible to progress by taking inspiration from biological systems (Danchin et al., 2004).

One view of cultural evolution is to recognise a correspondence between the processes underlying cultural and biological evolution: variation, reproduction, natural selection (Heylighen and Chielens, 2008). Rather than genetic transmission and recombination with variation as the mechanism of reproduction, cultural evolution considers the meme (Dawkins, 1976) as the unit of cultural transmission, i.e. communication. Memes may be transmitted and recognised by individual agents in the community. Variations may occur through errors in interpretation. Natural selection occurs since some memes are fitter than others: i.e., some memes are more likely to be communicated than others

(Heylighen, 1999). Thus, biology offers a framework for studying cultural evolution (Speel, 1995). However, cultural studies face the same problems as biological ones: it is impossible to measure everything and real-world complexity is overwhelming (Humphreys 2007).

To make progress, many social science experiments take a problem-led view of social behaviours, focusing on specific issues and building in assumptions about societal functioning to support analysis of the question posed, e.g., in emergent cooperation and communication (Buzing et al. 2005) and in language (Christiansen and Kirby, 2003). In Buzing et al. (2005), for example, results show that cooperation pressure leads to the evolution of communication skills that support cooperation. This cooperation pressure is built into the model, in that resource acquisition is directly enabled by cooperation. Communication is likewise built in, enabling recruitment of cooperators to acquire resource. Importantly, the model allows flexibility in the extent to which agents use communication – talking to request cooperation and listening to respond to cooperation requests – to interact with other agents. The work demonstrates the impact of environment (cooperation pressure) on communication strategy, and that the ability to listen occurs in advance of the ability to talk. Such a problem-led view thus focuses model construction on factors (measurables) and system dynamic assumptions that are likely to contribute to the phenomenon being investigated. While this approach limits the scope of the model to the question asked, it does provide insight into that question. Moreover, model results serve to refine the real-world question being set and direct iteratively the next phase of experimental design (Christiansen and Kirby, 2003) so focusing data collection on those measurables, and this in turn can refine the model construction (Bown et al., 2007).

Here, we take an alternative, complementary approach, where no assumptions are made about societal functioning and the goal is to elicit the fundamental processes responsible for the development of a proto-culture. This is similar in approach to Kirby (2001) where a protolanguage, lacking any structure, gives rise to a syntactic structure through evolution of the language itself rather than through evolution of the users of that language. Here, we outline an artificial culture laboratory designed that affords (re)production of memes, variation in meme production and a range of meme selection strategies. Through these fundamental processes, we are able to identify conditions that promote and inhibit both meme diversity and

reproductive fidelity. Our artificial culture lab comprises a physical arena with closed boundaries, populated by two-wheeled mobile robots called e-pucks, capable of moving forwards, moving backwards and turning (Mondada et al., 2009). They are equipped with a range of sensors that enable detection and tracking of obstacles and other robots. Importantly, robots can sense and track the movements of other robots nearby. Robots can signal to each other with movement and light (through programmable LEDs), and both movement and light may be detected through a simple on-board camera. Robots can also signal to each other through sound, as each has an on-board speaker and microphone. This allows multi-modal communication strategies on a one-to-one or one-to-many basis, and with or without active consent (i.e. one robot can eavesdrop on the communication between two others). The artificial culture lab is fully instrumented. A tracking system allows the movements of all robots to be captured and recorded for analysis and interpretation. The e-pucks have linux board upgrades (Liu & Winfield 2010).

We have implemented two modes of robot-to-robot communication: movement and sound. In each case, memes are the unit of cultural transmission. Our longer-term goal is to integrate movement-meme and sound-meme communication and investigate the evolution of multi-modal communication strategies. We have already published our movement and we detail sound-mediated communication here. For movement, memes are self-contained movement sequences (Winfield and Erbas, 2011). We refer to the robots as copybots (Blackmore, 1999) since they have no behaviours other than imitation, alternating from learner to teacher. While a teacher robot, seeded with one or more initial memes, enacts its meme, one or more learner robots observe that meme and store it in memory. When learner becomes teacher, a meme is selected from memory and enacted while other learner robot(s) observe. Importantly, we preclude robot-to-robot telepathy: the learner robot learns the meme enacted by the teacher through its senses alone. Consequently learners must solve the correspondence problem (Nehaniv and Dautenbaum, 2007), i.e. the problem of translating perceptions of another's actions (via sensory input) into corresponding motor actions.

The use of real physical robots, rather than simulated robots, together with the preclusion of robot-to-robot telepathy increases potential for emergence in behaviour, and in Winfield and Erbas (2011) we demonstrate that embodied movement-meme evolution is possible in the artificial culture lab. A combination of imperfect sensors, distance-dependent errors in sensor input and shared channels of communication provide a form of natural variation that drive novelty in the meme set. Specifically, artefacts emerging from this variation may give rise to new memes – and so new cultural “traditions” – that occur for no other reason than that they can (Winfield and Griffiths, 2010). For sound-memes, generated through e-puck speakers and heard through e-puck microphones, we also adopt the copybot concept. E-pucks move around an arena, listen to sound-memes sung by other e-pucks and then imitate what has been heard, under different meme selection strategies. Of note is that we were required to resort to simulation for our sound-meme experiments because

of practical limitations, particularly in regard to sound detection. With regard to sound generation, the e-pucks were not able to generate a consistent frequency. Moreover, the amplitude produced was very sensitive to battery level and so we had a very short operating window. With regard to sound detection, the e-pucks are very sensitive to: direction, needing to be directly facing the sound source under idealised (sound-proofed) conditions; distance, with a sharp distance-dependent attenuation; and ambient noise, with the noise created by e-puck movement being particularly problematic. The battery and directional constraints are removed in the simulator. Inconsistency in frequency of sound generation, i.e., natural variation, and the distance-dependent attenuation are accommodated in the simulator. We undertook systematic *in vitro* experiments to characterise that natural variation and then parameterise the simulator based on those experiments as in previous ecological studies (Bown et al. 2007). Ambient noise is eliminated by our use of simulation. The benefits of working with real robots are made clear in Winfield and Griffiths (2010), and the best compromise was to capture the natural variation through isolated robot-to-robot communication experiments and then develop software models of that variation. We are then able to integrate this sound-meme model into the real-world movement-meme robot laboratory to effect (a best approximation to) a system to explore the evolution of multi-modal communication.

## 2 Methods

### 2.1 Simulator Overview

The simulator was designed to simulate a number of e-puck robots moving around the Artificial Culture Lab, listening to and imitating each other's songs.

The simulator was designed to be a high-fidelity simulation of the transmission, detection and analysis of sound, and consequently we carefully calibrated the sound dynamics using real robots and these dynamics set the time step of the simulator (section 2.2). Our treatment of space and robot movement is of lower fidelity, yet designed to map onto the Artificial Culture Laboratory (section 2.3). In addition to the memory selection strategy described in Winfield and Erbas (2011), we explore alternate selection strategies (section 2.4) and the impact of these strategies on meme propagation in communities of robots (section 2.5).

### 2.2 Real-robot Calibration Experiments and Simulator Parameters

Extensive experiments were carried out to measure the sound signal generation and detection capabilities of the (real) e-pucks for simulator calibration. We identified a decay function for sound attenuation, the range of frequencies that could be both generated and detected by the e-pucks, a statistical model of variation in frequency generation, and upper and lower limits on speed of movement.

*Sound attenuation over distance* – the ability of e-pucks to

detect sounds generated by other e-pucks was measured over increasing distances. This is used to reduce the amplitude of the sounds over distance, ensuring that distant sounds are quieter and closer sounds are more readily detected.

*Inconsistent frequency* – e-pucks may be programmed to generate sounds at specific frequencies; however hardware limitations preclude generation of a clear, consistent tone. We measured variation in frequency generation over a range of frequencies (500Hz to 3200Hz). We then constructed a linear regression model to describe the average variation in for each frequency in that range in 100Hz intervals.

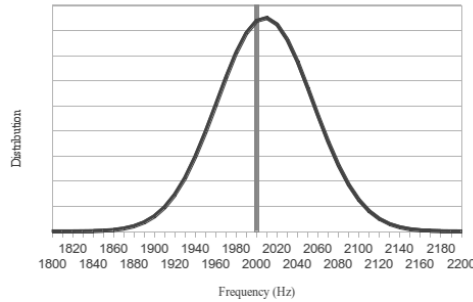


Figure 1 – Frequency Variation; Inset Sound Attenuation

Vertical line shows the target frequency of 2000Hz. The distribution curve shows the variation of the actual frequency generated by the e-Puck.

$$\text{MeanFrequency} = -16.8243 + (1.01247 * \text{TargetFrequency})$$

$$\text{StandardDeviation} = 46.9$$

The simulator samples from this distribution for every generated sound. Fig. 1 above shows the frequency distribution for target 2000Hz.

*Time-step and sound sampling rate* – e-pucks sample sound at 33kHz. Allowing for the (measured) time to transfer the samples from PIC to Linux processor (4ms) and 4ms processing time, e-pucks can process 128 samples every 8ms. The time-step in the simulator is set to 8ms. 128 samples gives the best ratio accuracy of frequency and timing measurement and gives a good range of usable frequency bands. More samples would decrease the timing measurement accuracy for little gain as the variation in frequency is greater than the increased accuracy of identified frequency ranges.

## 2.3 Simulation of Space and Memes

*The Simulated World* – the arena of the Artificial Culture Lab is represented by a lattice. The edge of the grid is a non-toroidal boundary. Each lattice cell represents an area approximating the size of an e-puck (5cm x 5cm). Each square may only be occupied by a single e-puck. Movement is represented by transitions from one square to an adjacent, unoccupied square, happening once every  $n$  milliseconds, where  $n$  is determined by the speed of the e-puck (note, the resulting simulated speed of the e-pucks is within the range of real e-puck speed). E-pucks move in straight lines, reflecting off lattice edges. When an e-puck attempts to move into a square occupied by another e-puck – i.e. a collision – it instead changes direction and moves to the first unoccupied

square in a clockwise direction, with respect to itself, from the occupied square, and if surrounded it does not move. The e-pucks are positioned centrally in the cell; consequently, for sound attenuation the distance between two e-pucks is equivalent to the distance between cell centres. E-pucks do not move if they are imitating or listening to other e-pucks. Note, a given e-puck may hear a single song, or multiple songs and this can result in interference.

*Memes* – a meme is, in the simulation, the representation of the song. Memes are mono-tonal, and the frequency that the meme is sang at is determined by the particular e-puck singing the meme. A meme is made up of a series of pulses of sound separated by periods of silence, and these are termed pulses of sound and pulses of silence respectively. The description of a meme, then, at its simplest is a list of pulse lengths (alternating periods of sound and silence) in milliseconds.

Here, the “idealised” form of a meme has been defined as a series of pulses of equal length. An idealised meme of three 250ms pulses of sound separated by 250ms pulses of silence would therefore be described as 250,250,250,250,250. Errors in transmission, detection or analysis of sound could result in a non-idealised meme described as 250,400,150,150,250.

*Memory: Distinct and Grouped* – to investigate different strategies of memory and selection, memory may be distinct or grouped. An e-puck with distinct memory will store every meme heard as a new meme, even if it is identical to a meme already in memory. An e-puck with grouped memory will examine every meme heard and determine if it is already known, in accordance with criteria defined below, or new. New memes are added to memory; already known memes have their count incremented.

*Memory: Short-term and Long-term:* - within the simulator the e-pucks have the equivalent of both long and short term memory. Short term memory stores the memes that have just been heard while long term memory stores all the memes that have been heard during the simulation run. While the e-puck is hearing sounds, those sounds are used to build up memes in the short term memory. When the e-puck hears silence for more than two seconds all the memes in short term memory are finalised, the e-puck decides which meme to sing next (from short or long term memory, depending on strategy), and copies all the memes from short term to long term memory.

## 2.4 Meme Description and Selection Strategies

*Meme* – a meme is a number of pulses of sound and silence. In its idealised form all pulses are the same length.

*Meme Metric* – we define a single metric comprising three measures: total meme length, in milliseconds; number of pulses in the meme; and a measure of the structural difference between the meme and an idealised meme of the same length and number of pulses.

$$\text{Structure} = \ln \left( \sqrt{\sum_k (\text{PulseLength}_k - \text{IdealisedPulseLength}_k)^2} \right)$$

Meme similarity is directly proportional to the Euclidean distance in the three-dimensional metric space.

Two memes are judged to be similar enough to be considered the 'same meme' if they have a) the same number of pulses, b) the same overall length to the nearest 500ms (one perfect pulse) and c) both structural measures are either below or above 6.214 (derived from perfect pulse).

*Selection Strategies* – here, we examine 4 different strategies for selecting which meme to imitate in response to a meme being heard.

1. *Random Mimicry* – after hearing a meme (or multiple simultaneous memes), add it to memory. When imitating, randomly pick a meme from that memory and sing it. This is a form of indirect mimicry. The e-puck does not mimic what it has just heard, it mimics something that it has heard at some point. Selection is weighted by how often the meme has been heard.

2. *Direct Short-term Mimicry* – the e-puck mimics one of the memes it has just heard, those which are in short term memory, picking randomly from those memes if more than one was heard. Selection is thus unaffected by the memes in long-term memory (which are stored for auditing).

3. *Direct Long-Term Mimicry* – the e-puck compares the memes it has just heard to the memes in its long term memory, determines which newly heard meme is most similar to one of the memes already in memory and mimics that heard meme. This is direct, memory-driven mimicry.

4. *Proto-Imitation* – the e-puck compares the memes it has just heard to the memes in its memory, determines which heard meme is most similar to one of the memes already in memory and sings the known meme from its memory. (most similar is closest in Euclidean distance in the three-dimensional metric space) A distinction can be made between mimicry, the copying of actions, and imitation, recognising the intent of those actions and enacting that intent. With this latter strategy the e-puck differentiates between the meme that it heard (the *action*) and the meme it believes the singer was trying to sing (the *intention*) and sings that intended meme in response. This is simple, proto-imitation rather than basic mimicry.

## 2.5 Experiments

The distance-dependent sound attenuation and inconsistency in frequency generation introduce potential differences between memes sung and memes heard. Moreover, when this is combined with e-pucks both singing and moving concurrently, new memes may emerge. New memes may arise from memes that: overlap, such that silences in one meme are filled with sounds of another; concatenate, since there are no special signals at the beginning or end of a meme it is possible to blend memes over time; and are generated with errors, through the model of inconsistency. The resulting complexities require us to first analyse meme evolution patterns in general (experiment 1). We also explore under what circumstances meme propagation is best effected (experiment 2).

*Experiment 1 – Meme evolution* – the impact of meme memory (distinct and grouped) and selection strategies on meme evolution was explored. The e-pucks were initialised with a set of four seeds in memory: i) five pulses of 300ms, ii) five pulses of 500ms, iii) five pulses of 700ms and iv) five pulses of 900ms. The other parameters varied across the tests were speed (eight speeds from stationary 0, to fastest 7) and population size (1 to 8 e-pucks). This resulted in 256 distinct tests, each of which was run multiple times.

We have devised “Memeographs” to report on these experiments. Our memeograph is a hierarchical graph that shows the connections between memes sung and memes heard. It identifies the individual robots involved, memes that are repeated and distinguishes between original seed memes and new memes. Nodes are memes and links are “listening events”. For a given link, the meme at the arrow end was heard (and added to memory) as a result of hearing (or mishearing) the meme at the tail end. While the memeograph contains time-based information, it should not be misinterpreted as a time-line; in particular, a chain of memes should not be taken as evidence that a meme was sung multiple times in a row – it is possible that that chain was sung intermittently with other memes being sung in between. The colour of the node indicates the meme. Node shape indicates the robot. Seed memes are depicted larger than new memes.

*Experiment 2 – Meme Propagation* – We examine the propagation of a single meme from a single e-puck across a community. Initially, one e-puck knows a single meme; seven other e-pucks begin with no memes in memory. Consequently, these e-pucks construct their memories from this seed meme only when they encounter and hear that meme, or some corrupted form of that meme. We analyse under varying conditions the rate of meme propagation. The experiments were replicated fifty times for each memory / selection strategy pairing at eight different speeds (0 to 7). We varied seed meme complexity: the shortest meme consisted of three 500ms pulses (sound, silence, sound); other seed memes were two pulses longer (5, 7, 9, 11 13, 15 and 17 pulses).

## 3 Results

### 3.1 Meme Evolution

Memeograph topology is predominantly affected by memory strategy and selection strategy. Speed had no effect on diversity, except for zero speed which precluded meme exchange and therefore evolution. Diversity scales linearly with population size (results not shown). For illustration we show four of the possible eight combinations below (Fig. 2). For each memory and selection strategy combination, we use the Random Mimicry selection strategy as a baseline to express the average number of memes and clusters of memes, determined by QT Cluster Analysis (Heyer et Al, 1999). The QT Cluster Analysis is based on the metric described in 2.4, normalised to a 0.0 to 1.0 range, with a radius of 0.05 and a

minimum cluster size of five.

	Memory Strategy			
	Grouped		Distinct	
Selection Strategy	Clusters	Memes	Clusters	Memes
1. Random Mimicry	1	1	1	1
2. Short-term Direct Mimicry	0.91	1.02	0.64	0.90
3. Long-term Direct Mimicry	0.94	1.0	0.66	0.80
4. Proto-Imitation	0.98	0.98	0.75	0.81

Table 1 - Meme Evolution Results

Table 1 shows the relative numbers of memes and clusters (groups of memes) by selection and memory strategies

**Grouped Memory** - Memeographs of grouped memory experiments indicate how often a given link has happened with a number on the link and by the thickness of the link line. Grouped memory results in fewer memes in memory and fewer clusters than Distinct memory. Table 1 above shows the relative sizes of the number of memes and groups of memes generated by the different memory and selection strategies.

1. *Random Mimicry* – This is the base line case for Grouped Memory strategy, it produces more clusters and memes than any other selection strategy (when using grouped memory).
2. *Direct Short Term Mimicry* – (Fig. 2b). For the grouped memory strategy, the Direct Short Term Mimicry selection strategy produces the least clusters of all the selection strategies. This is because the mimicry is only based on the memes most recently heard, limiting the number of memes that could be sung at any point. There is an increasing chance that the memes just heard will already be in memory.
3. *Direct Long Term Mimicry* - Comparing the memes heard to the memes in memory has the effect of increasing the number of memes involved in the mimicry, resulting in more diversity of memes than the short term mimicry.
4. *Proto-Imitation* - results in slightly fewer clusters and memes than the Random Mimicry strategy. Singing a meme heard earlier rather than the meme just heard increases the likelihood that the meme sung will be different to the heard meme.

**Distinct Memory** – With every meme stored as a distinct meme this strategy produces 47% more clusters and 755% more memes than the grouped memory strategy.

1. *Random Mimicry* – all memes in memory are equally likely to be chosen, and over time most memes will be sung multiple times. This is the base

line case for Distinct Memory strategy.

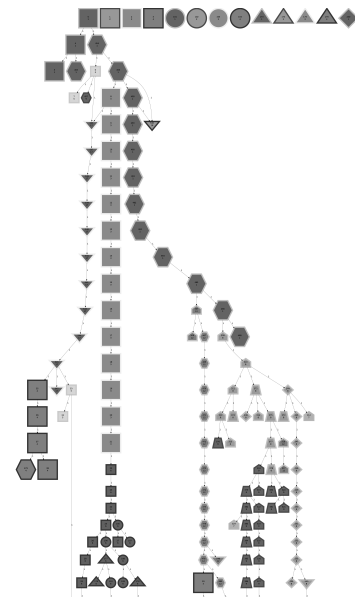


Figure 2a – Direct Memory, Direct Short-term Mimicry

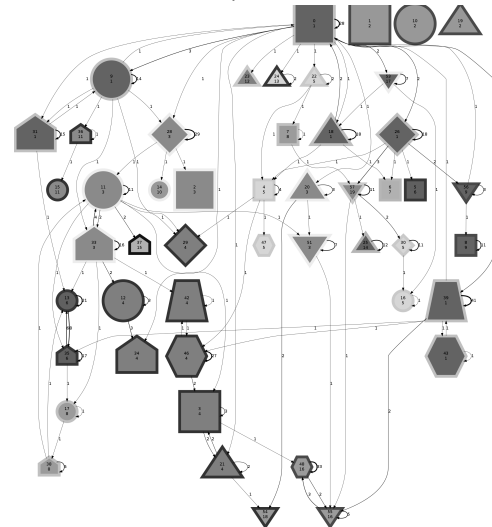


Figure 2b – Grouped Memory, Direct Short-term Mimicry

Each node identifies a meme in the memory of an e-puck. The shape of the node identifies the e-puck, the colour identifies the meme and the size distinguishes between seed memes (large) and emergent memes (small).

2. *Direct Short Term Mimicry* – Since each meme heard is stored as distinct meme in memory, long chains of nodes with the same shape and colour occur (Fig. 2a). This combination reduces the number of clusters more than any other strategy but does not reduce the overall number of memes as much as Direct Long Term Mimicry and Proto-Imitation.
3. *Direct Long Term Mimicry* – results in almost as low a number of clusters as Direct Short Term Mimicry but with far fewer memes stored.



4. *Proto-Imitation* – this too is a very limiting strategy though the effect on the memeograph is completely different. With this strategy, only seed memes are ever sung. Whatever the listener hears it interprets it as a prompt to sing a seed meme, as such no newly heard meme is ever more than one step away from a seed meme.

Using Random Mimicry as a baseline the other strategies reduce the number of clusters found for both memory strategies, with a larger reduction for the distinct memory strategy. They do not change the overall number of memes generated with the grouped memory strategy as they do with the distinct memory strategy. This is interpreted as the selection strategies having the effect of implicitly grouping the memes, explaining why this effect of the selection strategies is greater when the memory strategy does not group the memes (i.e. Distinct Memory).

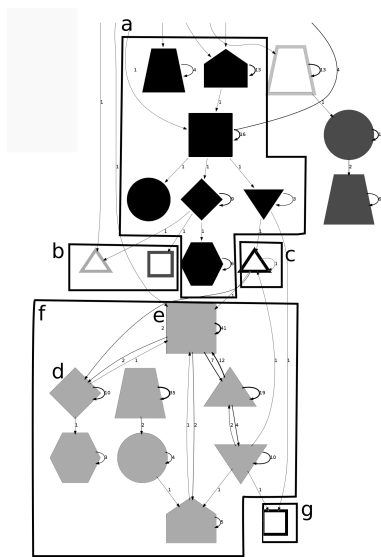


Figure 3 – Memeograph Detail – Grouped Memory

Each node identifies a meme in the memory of an e-puck. The shape of the node identifies the e-puck, the colour identifies the meme and the size distinguishes between seed memes (large) and emergent memes (small).

a & f – two distinct memes, each known by multiple e-pucks

b, c & g – emergent memes created by incorrect mimicry of memes

d & e – two different e-pucks incorrectly mimicking meme c

A closer inspection of the memeograph affords an account of experiment dynamics, as depicted in the fragment shown in Fig. 3. In area a, the large shapes show a seed meme is mimicked successfully (nodes have same colour) by four robots (different shapes). Examination of the whole memeograph (not shown) reveals that meme is a stable meme that is sung and heard correctly often.

However it is not mimicked correctly every time: the three small shapes in areas b and c are coloured differently indicating that either the imitator sang the meme badly, or that the listener misheard, perhaps by not hearing the start of the meme.

Interestingly the small triangle (area c) was heard by two

other robots, diamond (d) and square (e), both of which misheard the meme in the same way, as shown by them having the same colour. More interesting still, this misheard meme proves to be a stable successful meme itself and gets passed on to all the other robots in the experiment (area f).

Selection strategies of Random Mimicry and Long-term Mimicry have more diversity than those of Short-term Mimicry and Proto-imitation (Fig. 4). Speed and meme complexity have no effect on diversity.

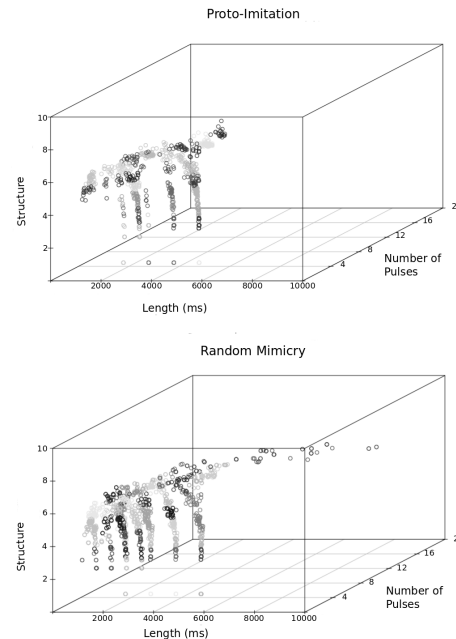


Figure 4 – Homogeneity and Diversity

Two 3d scatterplots of memes, each meme is plotted in three-dimensional space by its metric; the x-axis is the length of the meme in milliseconds, the y-axis is the number of pulses in the meme and the z-axis is the measure of distortion from the idealised form of the meme. The memes are coloured by similarity derived from QT cluster analysis of the data.

### 3.2 Meme Propagation

Fig. 5 shows the average spread across the community of initial seed memes of differing complexity for each of the memory/selection strategy pairs. The x axis shows the speed of the e-pucks. The y axis is number of robots the seed meme has spread to, varying from zero (no propagation) to seven (maximum propagation). As seed meme complexity increases meme propagation decreases: a shorter meme is more likely to be spread throughout the community since there is less chance of making a mistake in a shorter meme or of only hearing part of it.

Memory strategy has little effect on the propagation of the seed meme. In contrast the choice of selection strategy has a marked effect.

*Random Mimicry* – proves to be an effective strategy to propagate the seed meme. It ensures that if a robot has correctly heard the seed meme it has a chance to repeat it (or any other meme) at some point.

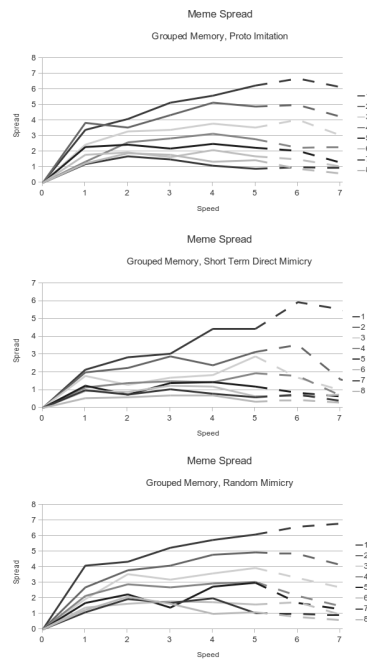


Figure 5 – Meme spread by meme length and e-puck speed

Speed varies from 0 (stationary) to 8 (fastest). Memes vary from shortest (1) to longest (8). Spread varies from 0 to 7 (full propagation). Dashed lines above speed 5 indicate speeds the real e-pucks are incapable of.

*Direct Short Term Mimicry* is the least successful at spreading the seed meme. Failing to compare what is heard to what is in memory means that any drift away from the seed meme will only be corrected by chance errors in communication.

*Direct Long Term Mimicry* is a more effective strategy. The more often a meme has been heard the more likely it is to be mimicked, and an error by a single e-puck is likely to be ignored. However by mimicking the heard meme that is closest to a known meme there will inevitably be some incremental drift away from the seed meme. This drift may lead to a meme that is greatly different from the seed meme.

*Proto-Imitation* is another effective strategy. The more often a meme is heard the more likely it is to be repeated. By repeating the known meme that is closest to the heard meme rather than mimicking the heard meme any errors in the heard meme are ignored and the correct version of the meme is sang. This limits the incremental drift problem.

Speed has a significant effect on the spread of the seed meme. If the community is moving too slowly there is insufficient opportunity for meme exchange. For experiments that replicate speeds at which the e-pucks are capable of moving (speeds 1-5) we see that as speed increases there is increased spread of the seed meme, with the greatest increase in spread at the lower speeds and a plateau-effect at higher speeds. To investigate this plateau we simulated the effects of robots that could move faster than the e-pucks (speeds 6+), and observed that at higher speeds meme spread reduces. E-pucks moving

at high speeds are typically hearing only part of memes being sung as they move away from neighbours too quickly. Very short memes (meme 1 and to some extent meme 2) are less affected by the high speed drop off.

## 4 Discussion

We explored sound meme transmission in a simulated community of e-Puck robots. We identified conditions that promote and inhibit both meme diversity in and reproductive fidelity of sound memes in the artificial culture laboratory.

### 4.1 Meme Evolution

The grouped memory strategy, i.e. only storing distinctly different memes as new memes and recording how many times each meme had been heard, resulted in fewer memes than distinct memory, i.e. storing every meme heard as a distinct meme. Grouped memory also results in fewer meme clusters and therefore a more homogeneous set of memes.

Grouping memes eliminates small mutations in the memes that the distinct memory approach keeps. Consequently, meme evolution occurring in small steps through multiple imitations (iterations) is precluded. As a result, changes that are observed occur from larger meme mutations, resulting in distinctive memes occurring at much lower frequency.

Short-term direct mimicry, i.e. mimicking one of the memes that has just been heard without any reference to other memes in long term memory is a conservative strategy. Compared to the other selection strategies, this results in the fewest clusters and therefore less diversity. If, at any point, a single meme is more commonly sang than any other then it is likely to remain the most common meme sang (since the chance of randomly selecting it increases as more e-pucks sing it). A very different meme is therefore less likely to be mimicked and one off errors in singing or listening will have little effect on the system.

In contrast, random mimicry maximises diversity in memes. The random responses greatly reduce the chance that the e-pucks will synchronise on a single meme, and allows one off memes that the other strategies would ignore a chance to propagate.

The effect of allowing long-term memory to influence imitation, through long term direct mimicry or proto-imitation, is to reduce diversity. Short-term mimicry is driven by the pattern of the most recently heard memes, increasing the chance that the e-pucks will synchronise the memes they are singing, resulting in fewer distinct memes and more homogeneity. Long term memory does though have a greater effect on reducing the overall number of memes in the system. To maximise diversity, a combined strategy of distinct memory with random mimicry is best. To minimise diversity, a strategy of grouped memory with short-term direct mimicry is optimal. Note, diversity scales linearly with population size and not at all with speed, apart from speed zero which limits exchange and effectively precludes meme evolution.

## 4.2 Propagation

In addition to the impact of meme selection strategy, detailed at length in section 3.4, the successful propagation of specific memes among the e-Puck population varies in accordance with meme length, meme complexity and speed of movement. Meme length and complexity impact successful meme propagation. Short, simple memes propagate through the community more successfully than longer, more complex ones and are less likely to be truncated or misheard, and during a fixed time period, shorter memes can be repeated more often and are more likely to be heard throughout the community.

Speed of movement also affects meme propagation. If the e-Pucks are not moving at all then there is limited or no opportunity for propagation: any e-Puck out of ear-shot from the others will never receive the meme. For any but the shortest memes, moving too quickly has a limiting effect on propagation, fast e-Pucks seldom hear the full meme.

In summary, to propagate a specific meme the best approach is for a simple meme with the robot moving at an average speed using any strategy except short term mimicry. Moreover, to propagate a specific meme while keeping diversity as low as possible - and so to maximise the impact of that seed meme - the grouped memory and long term, direct mimicry strategies should be used.

## 4.3 Conclusions

We focused on sound memes and used a similar experimental approach to our work on movement-memes (Winfield and Erbas, 2011), which reported embodied movement-meme evolution. Because of limitations in the e-Puck platform, we had to recourse to simulation but we encapsulated the observed natural variation in e-Pucks through model parameters. Here, as in Winfield and Erbas (2011), we demonstrate that meme selection strategy, when combined with natural variation and reproduction through imitation, is a crucial factor in cultural evolution.

Much progress has been made in evolutionary biology and we, like Mesoudi et al. (2006), believe that much progress can be made in cultural evolution by adopting methodologies from biology. We propose that our artificial culture lab provides a real-world framework with natural variation for controlled experimental simulations in embodied, multi-modal cultural evolution. We can explore meme-gene coevolution (Bull et al., 2000), the influence of environmental variation on selection (Kingsolver et al., 2003), and the links between micro- and macro-evolution scales. Kline and Boyd (2010) note that larger populations generate more complex cultural adaptations than smaller, isolated ones. In their review they indicate that chance events that perturb cultural transmission are more impacting in small populations. Moreover, errors in transmission will cause complex traits to degrade more quickly than simple traits, although large populations mitigate this. As a complement to such empirical studies, our experimental framework allows exploration, in a controlled way, of group selection processes (Boyd and Richerson, 2010) in the context of individual variation. We can thus investigate

the relation between individual decision-making and community-scale cultural phenomena.

**Acknowledgements** - This work is supported by the UK Engineering and Physical Sciences Research Council (EPSRC), grant reference EP/E062083/1, and the authors gratefully acknowledge project co-investigators and researchers.

## References

- Blackmore, S. (1999) *The Meme Machine*, Oxford University Press
- Bown, J.L., Pachepsky, E., Eberst, E., Bausenwein, U., Millard, P., Squire, G.R., Crawford, J.W. (2007). Consequences of intraspecific variation for the structure and function of ecological communities Part 1: Model development and predicted patterns of diversity. *Ecological Modelling* 207, 264-276.
- Boyd, R. and P. J. Richerson. Transmission coupling mechanisms: cultural group selection. *Philosophical Transactions of the Royal Society (B)*, 365, 3787-3795, 2010.
- Bull, L., Holland, O. & Blackmore, S. (2000). On Meme-Gene Coevolution. *Artificial Life* 6(3): 227-235.
- Buzing, P.C., Eiben A.E. and Schut, M.C. (2005) Emerging Communication and Cooperation in Evolving Agent Societies. *Journal of Artificial Societies and Social Simulation* 8(1).
- Christiansen, M.H. and Kirby, S. (2003). Language evolution: consensus and controversies. *TRENDS in Cognitive Sciences*, 7(7).
- Danchin, E., Giraldeau, L., Valone, T. and Wagner, R. 2004. Public information: from nosy neighbors to cultural evolution. *Science*, 305, 487-491.
- Dawkins, R. (1976) *The Selfish Gene*, Oxford University Press
- Heyer LS, Kruglyak S, Yooseph S (1999) Exploring expression data: Identification and analysis of coexpressed genes. *Genome Research*, 9:1106-1115.
- Heylighen, F. (1999). What makes a meme successful? Selection criteria for cultural evolution in *PRoc. 15th Int. Congress on Cybernetics*, p 418-423.
- Heylighen F. and Chielens, K. (2008) *Cultural Evolution and Memetics Encyclopedia of Complexity and System Science*
- Humphreys, P. (2007). *Extending Ourselves: Computational Science, Empiricism, and Scientific Method*. Oxford University Press, USA.
- Kirby, S. (2001) Spontaneous Evolution of Linguistic Structure - An Iterated Learning Model of the Emergence of Regularity and Irregularity. *IEEE Transactions on Evolutionary Computation*, 5(2).
- Kingsolver, J. G., and R. Gomulkiewicz. 2003. Environmental variation and selection on performance curves. *Integr. Comp. Biol.* 43:470-477.
- Kline, M. and R. Boyd, Population size predicts technological complexity in Oceania, *Proceedings of the Royal Society (B)*, 277, 2559-2564, 2010.
- Liu W, Winfield AFT (2010), 'Open-hardware e-puck Linux extension board for experimental swarm robotics research', *Microprocessors and Microsystems* 35 (1), 2011, doi:10.1016/j.micpro.2010.08.002.
- Mesoudi et al. (2006), Towards a unified science of cultural evolution. 2006. *Behavioural and Brain Sciences*. 29, 329-383.
- Mondada, F., Bonami, F et al., (2009). The e-Puck, a Robot Designed for Education in Engineering. Pages 56-65 of 9th Conference on Autonomous Robot Systems and Competitions, vol. 1 Portugal.
- Nehaniv, C. and Dautenhahn, K. (eds) (2007). *Imitation and Social Learning in Robots, Humans and Animals*. Cambridge University Press.
- Speel, H.-C. (1995), *Memetics: On a conceptual framework for cultural evolution*. in Symposium: Einstein Meets Margritte: Free University of Brussels
- Winfield AFT and Griffiths F (2010), 'Towards the Emergence of Artificial Culture in Collective Robot Systems', pp 431-439 in *Symbiotic Multi-robot Organisms*, Eds. P Levi and S Kernbach, Springer-Verlag,
- Winfield AFT and Erbas M. (2011) On embodied memetic evolution and the emergence of behavioural traditions in robots. Accepted for *Journal of Memetic Computing*.

# Embodied swarming based on back propagation through time shows water-crossing, hourglass and logic-gate behaviors

Yukio-Pegio Gunji<sup>1,5</sup>, Hisashi Murakami<sup>1</sup>, Takayuki Niizato<sup>1</sup>, Andrew Adamatzky<sup>2</sup>, Yuta Nishiyama<sup>1</sup>, Koichiro Enomoto<sup>3</sup>, Masashi Toda<sup>3</sup>, Toru Moriyama<sup>4</sup>, Tetsuya Matsui<sup>1</sup> and Kojiro Iizuka<sup>4</sup>

<sup>1</sup>Kobe University, Department of Earth & Planetary Sciences, Kobe University, Kobe 657-8501, JAPAN, <sup>2</sup>University of the West England, Bristol, BS16 1QY, UK <sup>3</sup>Hakodate Future University, Hakodate 041-8655, JAPAN, and <sup>4</sup>Young Researchers Empowerment Project, Shinshu University, Ueda 386-8567, JAPAN

<sup>5</sup>Corresponding author: yukio@kobe-u.ac.jp

## Abstract

A flock, school, and swarm are collective behaviors that can be compared to a human consciousness or body. Through recent developments in image analysis and model simulation, it has been found that the collective behavior of animals can, as a whole, show characteristics of a single “body”. It has also been found that intrinsic noise can positively contribute to swarming and/or flocking. Motivated by field observations of soldier crabs, *Mictyris guinotae*, we propose a swarm model based on inherent noise and back propagation in time that mimics mutual anticipation. A swarm generated by this model is characterized by flexible, dynamical and robust behavior containing inherent turbulence. We demonstrate that the model can produce water-crossing, hourglass and logic gate behaviors, which are also found in real soldier crabs. We describe how a sense of ownership and a sense of agency of the “body” arise in our model, and we propose that the concept of a body should be verified in terms not of stability but of robustness.

## Introduction

Does a swarm, flock or school have a single consciousness or body (Vicsek, 2001, Couzin, 2007; 2008; 2010, Sumpter, 2010)? This question has been addressed in the context of collective decision making by computer models, particularly BOIDS (Reynolds, 1987) and SPP (Vicsek et al., 1995, Czirók et al. 1996). Owing to developments in image analysis that have made it possible to obtain kinetic data on the movements of real organisms (Ballerini et al., 2008a, b, Carere et al., 2009), several internal dynamical structures within groups have recently been identified. These structures include topological distance (Ballerini et al., 2008b), scale free correlation (Cavagna et al., 2010) and inherent noise (Yates et al., 2009). This research also suggests that inherent turbulence could play an essential role in collective motion. The collective behaviors of animals might be based on inherent noise, and the internal structures of a group are perpetually generated and modified to maintain a robust unity as a whole.

A flexible but robust swarm (flock or school) can be compared to a human’s body (Gunji et al., 2010). In this sense, an animal group might recognize external objects in the

environment by an embodied cognitive process (Varela et al., 1992, Pheifer and Scheier, 2001, Pheifer et al., 2007). Human body awareness can be described by a sense of ownership (i.e., the sense that I am the one who is undergoing an experience) and of a sense of agency (the sense that we are the initiators of our actions) (Wegner et al., 2004, Tsakiris, et al., 2008). Although a body appears to be very stable and unambiguous, it is well known that synchronous visuo-tactile stimulus can make body illusions, such as the rubber hand illusion (Botvinick, M. and Cohen, 1998) and an out-of-body experience (Lenggenhager et al., 2007, Ehrsson, 2007) possible. The body is also a robust and flexible system that can be adapted to environments. The problem still remains whether a swarm, flock and school can be compared to a “body” in these senses.

Here, we show how inherent noise in conjunction with organisms’ mutual anticipation can actively contribute to the generation and maintenance of a robust swarm in a computer model. Mutual anticipation was implemented by asynchronous updating and back propagation through time. The time slice of a swarm is thus so complex that a swarm is robustly maintained and contains inherent turbulence. The model was constructed through observations of soldier crabs, *Mictyris guinotae* (Bradshaw and Scoffin, 1999, Shih., 1995, Peter et al., 2010). Our model can reproduce a swarm entering and crossing water through the emergence of a highly concentrated subpopulation driven by inherent turbulence; an hourglass of crabs showing regular oscillations; and collision-based-computing logic gates implemented by a swarm ball. The generation of these behaviors depends on the robustness and flexibility of swarming. Finally, we argue that a body-like character is embedded in our swarm model in the form of the interplay between anticipation and memory.

## Swarming by mutual anticipation

Through observations of soldier crabs in the Iriomote Islands, Okinawa Prefecture, Japan, we discovered a role for inherent turbulence in collective behavior. A swarm of soldier crabs always contains inherent turbulence such that individuals in a swarm have different velocities, while the swarm maintains a coherent and dense unity. Inherent

turbulence in particular plays an essential role in water-crossing behavior. When a small swarm of soldier crabs confronts a water front, it cannot enter the water and moves along the perimeter of the water pool. In moving along the water front, inherent turbulence creates a highly concentrated locus inside the swarm, which can then enter and cross the water.

If inherent turbulence provides the essential mechanism to generate robust collective behavior, an important question is whether this robustness can be distinguished from stability. In the context of stability, perturbations conflict with the mechanism that generates order. In the context of robustness, inherent noise positively contributes to the generation of collective behavior. To implement inherent noise, we proposed a mechanism of mutual anticipation based on multiple potential transitions.

### Basic model

A model is defined in a discrete space of  $S \times S$  with  $S = \{1, 2, \dots, S_{\text{MAX}}\}$ . The co-ordinate of the  $k^{\text{th}}$  agent at the  $t^{\text{th}}$  step is given by

$$P(k, t) = (x, y) \quad (1)$$

where  $x \in S, y \in S$ , and  $k \in K = \{1, 2, \dots, N\}$ . Each  $k^{\text{th}}$  agent at the  $t^{\text{th}}$  step has  $P$  number of potential vectors  $v(k, t; i)$  with  $i \in I = \{0, 1, \dots, P-1\}$ . If  $i = 0$ , the vector is  $v(k, t; 0)$ , which is known as the principal vector. Otherwise, the vector is represented by the angle  $\theta_{k,t}$ , such that

$$v(k, t; i) = (\text{Int}(L\eta_i \cos(\theta_{k,t} + \xi_i)), \text{Int}(L\eta_i \sin(\theta_{k,t} + \xi_i))) \quad (2)$$

where for any real number  $x$ ,  $\text{Int}(x)$  represents integer  $X$  such that  $X \leq x < X + 1$ .  $L$  is the length of principal vector. Because of the wrapped boundary,  $X$  belongs to  $S$ . If  $i \neq 0$ , random values  $\eta_i$  and  $\xi_i$  are selected with equal probability from  $[0, 1]$  and  $[-\alpha\pi, \alpha\pi]$ , respectively. The target of the vector is represented by  $\alpha(k, t; i) = P(k, t) + v(k, t; i)$ .

The mutual anticipation depends on the popularity of a site,

$$\zeta(x, y; t) = \begin{cases} |\{\alpha(k, t; i), k \in K, i \in I \mid \alpha(k, t; i) = (x, y)\}|, & \text{If } \forall (k \in K) P(k, t) \neq (x, y); \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

$\zeta(x, y; t)$  represents the number of potential transitions whose targets reach the site  $(x, y)$  where there is no agent. Before updating the location, for any  $(x, y)$  at the  $t^{\text{th}}$  step and  $\omega(x, y; t) \in \{0, 1\}$  we set  $\omega(x, y; t) = 0$ .

The agents' locations are updated asynchronously. If there exists  $i \in I$  such that  $\zeta(\alpha(k, t; i)) \geq 2$ , the next site for the  $k^{\text{th}}$  agent is defined by

$$P(k, t+1) = \alpha(k, t; s), \quad (4)$$

where  $s$  satisfies the condition such that for any  $i \in I$ ,  $\zeta(\alpha(k, t; s)) \geq \zeta(\alpha(k, t; i))$ . These conditions ensure that an agent moves to the most popular site. If multiple sites satisfy this condition, one is chosen randomly. Because the popularities are propagated backward in time, agents in a swarm can anticipate each other's moves.

A set of updated sites is represented by  $U_N = \{(x, y) \in S \times S \mid P(k, t+1) = (x, y)\}$ . The vacated site generated by equation (4) is recorded in memory as  $\omega(x, y; t) = 1$  if  $P(k, t) = (x, y)$  and  $P(k, t+1) \in U_N$ . An agent that is not updated by equation (4) then moves to the vacated site by

$$P(k, t+1) = \text{Rd}\{(x, y) \in N_f \mid \omega(x, y; t) = 1\}, \quad (5)$$

if  $|\{(x, y) \in N_f \mid \omega(x, y; t) = 1\}| \geq 1$ , where  $\text{Rd}J$  represents an element randomly chosen from set  $J$ , and  $N_f$  is the follower's neighborhood. The agent moving by eq-(5) is called a follower because it follows a predecessor.

If an agent is not updated by (4) or (5), it moves by

$$P(k, t+1) = \text{Rd}\{\alpha(k, t; i) \mid (j \in K') P(j, t) \neq \alpha(k, t; i) \wedge \alpha(k, t; i) \notin U_N\} \quad (6)$$

where  $K'$  is an index set of agents that are not updated. An agent moving by eq-(6) is called a free mover.

Finally, principal vectors are locally matched with each other in the neighborhood through velocity matching,  $M$ . This matching operation is expressed as

$$\theta_{k,t+1} = \langle \theta_{k,t} \rangle_M. \quad (7)$$

The bracket with  $M$  represents the operation of averaging velocity directions in the neighborhood,  $M$ .

Fig. 1a shows the neighborhood of velocity matching and of the follower. Figure 1b shows the procedure of velocity matching, mutual anticipation, and following.

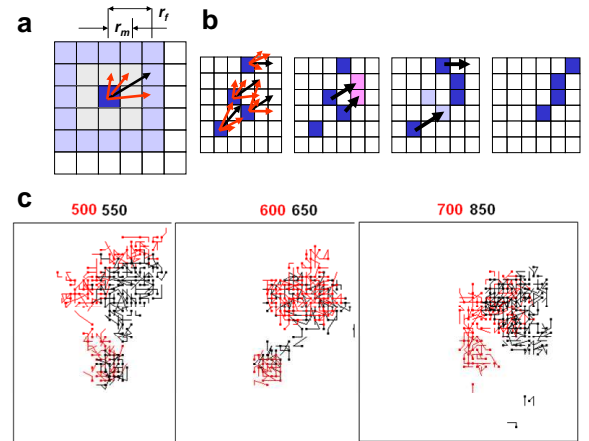


Figure 1 Schematic diagram of the transition and time development of the model simulation: (a) Principal vector (black arrow) and alternative vectors (red arrows) of a crab (blue square) in the matching neighborhood with radii  $r_m$  (pale gray lattices) and the following neighborhood with radii  $r_f$  (pale gray + pale blue lattices). (b) Transitions of crabs in a two-dimensional discrete space. Velocity matching (far left), mutual anticipation (second from left), following and free movement (second from right) and the resulting distribution at the next step (far right). (c) Time development ( $t = 500-850$ ) of our swarm model of 100 agents, with  $P = 20$ ,  $\alpha = 0.9$ ,  $L = 4$ ,  $r_a = r_f = 2$ .

First, velocity matching is applied to the principal vectors, and the agents then move to the most popular site (pink site in Fig. 1b), yielding a vacant site (pale blue site in Fig. 1b). Highly popular anticipated sites propagate backward in time, revealing the asynchronous transitions. Thus, mutual anticipation is here implemented by back propagation in time. Agents move to a vacant site if it is within the follower's neighborhood. Fig. 1c shows a series of snapshots of our swarm model. Each agent is represented with its 5-step trajectories. It is easy to see that a swarm contains turbulent motion despite maintaining a highly dense whole.

Fig. 2a shows polarization/density of a swarm plotted against external perturbation in our model with  $P = 1$ . Polarization is defined by the length of the average velocity over all agents in a swarm. Density is defined by the average number of agents in the neighborhood of  $20 \times 20$  lattices. In the model with  $P = 1$ , the external perturbation,  $\xi$ , is randomly chosen from  $[0, 1]$  and is coupled with velocity matching. When the projected velocity of agent is expressed as  $(v_x, v_y)$ ,  $v_x + \xi$  and  $v_y + \xi$  are given for the unit vector. If  $P = 1$ , the model corresponds to BOIDS because each agent has only one velocity. The coherence of a swarm can result only from velocity matching or high polarization; the more polarized and dense the population, the less external perturbation there is.

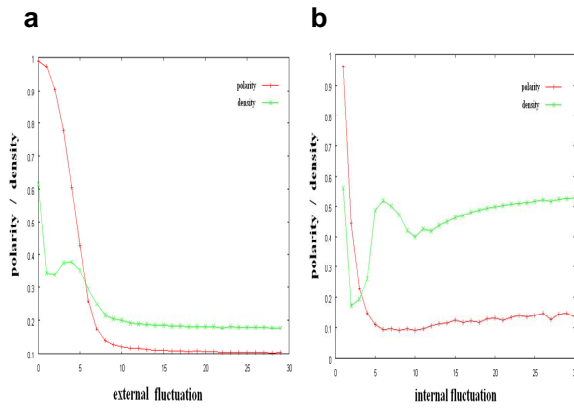


Figure 2 Polarization/density plotted against perturbation. (a) The polarization/density ratio plotted against external perturbation in the model with  $P = 1$ . (b) The ratio plotted against internal perturbation, which is defined as the number of potential transitions normalized by the maximum number of potential transitions. For this plot,  $P$  ranged from 1 to 30.

Fig. 2b shows the polarization/density of a swarm plotted against the inherent perturbation in our model. The inherent perturbation is expressed by  $(P - 1)/P_{\text{MAX}}$ , where  $P$  is given from 1 to  $P_{\text{MAX}}$  (30). The more inherent noise (i.e., more  $P$ ) that is present, the higher the density and the lower the polarization are. This relationship reveals that a highly dense swarm is generated by mutual anticipation and/or inherent noise. For this reason, a coherent swarm (i.e., a highly dense swarm with an extrinsic boundary) contains inherent turbulence.

In the next section, we illustrate how inherent noise positively contributes to robust swarm behavior by demonstrating the role of noise in water-crossing behavior, hour glass behavior and collision-based computing implemented by swarm balls.

### Water-crossing behavior

The water-crossing behavior observed in real soldier crabs can be easily approximated by our model. To introduce a tidal pool into the simulation, we define a specific area  $U_p \subseteq S \times S$  in which the condition allowing mutual anticipation is replaced by

$$\zeta(\alpha(k, t; i)) \geq c. \quad (8)$$

The value  $c$  is an integer larger than 2. Because  $c > 2$ , it is more difficult for agents to go through the area  $U_p$ . Only by introducing the specific area  $U_p$  can we simulate the behavior of crossing water.

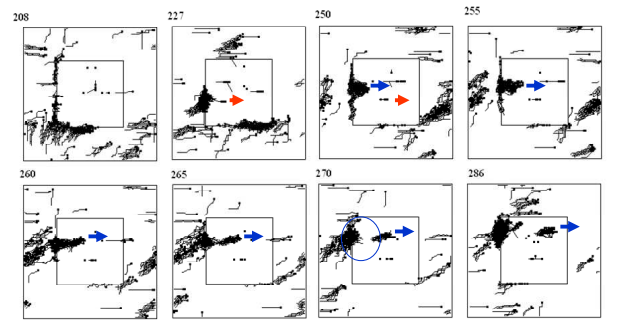


Figure 3 Snapshots of the time development of swarm trajectories in a model simulation. Numbers in the upper left of each plot denote the time step. Each agent is represented with its 5-step trajectories. The rectangle located in the center indicates  $U_p$ , which represents a tidal pool. For these simulations,  $P = 10$ ,  $\alpha = 0.3$ ,  $L = 4$ , and  $r_a = r_f = 2$ . Blue and red arrows represent the directions of motion of swarms. Blue circle represents highly concentrated area of a swarm.

Fig. 3 shows a series of snapshots of our swarm model demonstrating water-crossing behavior. Although a single agent or a small swarm cannot enter the tidal pool, a highly concentrated, large swarm can enter and cross the tidal pool. These behaviors are consistent with the behaviors of real soldier crabs observed in Iriomote islands.

Fig. 4 shows the frequency of a swarm invading the tidal pool in a model given by  $P = 20$ ,  $\alpha = 0.5$ ,  $L = 4$ , and  $r_a = r_f = 2$ . For each experiment, we prepared 2000 cases of a swarm confronting the tidal pool. If the size of a swarm exceeds a certain value, a constant high probability of invasion is achieved. If  $P$  is smaller, the possibility of which the popularity exceeds the threshold decreases. Thus, the minimum size of a swarm invading the tidal pool increases.

Because a swarm generated by our model is so robust that a swarm can go through the tidal pool once the swarm enters

the water. Yet, if an agent is isolated in the water, then he or she cannot move and is left alone. This phenomenon is also observed in real soldier crabs.

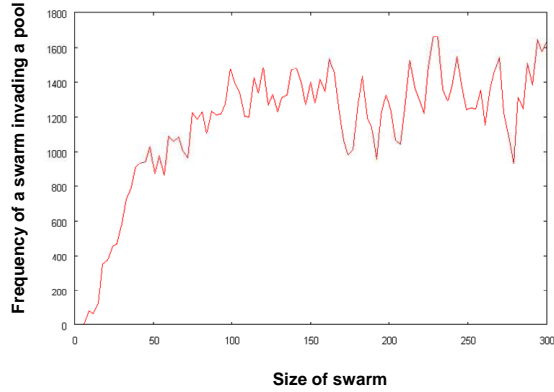


Figure 4 Frequency of a swarm invading a tidal pool as a function of swarm size.

### Hourglass made of soldier crabs

In field observations, we found a soldier crab moving along the wall in a closed container, and we created an hourglass made of real soldier crabs (Nishiyama et al., 2011). Forty soldier crabs are collected and are confronted in a closed container, where the floor was made of cork providing friction enough to walk for soldier crabs. If the container is left for while, soldier crabs walk along the wall in keeping a half-broken swarm. Since a concentrated swarm oscillates along the wall, the hourglass made of soldier crabs produced periodic oscillations for approximately two or three hours with a period of 70 seconds.

This behavior can be approximated by our model by slightly modifying one rule.

To simulate hourglass behavior as shown in Fig. 5 and 6, we introduced a tendency to walk along a wall. The hourglass scenario is constructed as follows. We first defined the wall state for any lattice  $(x, y)$  such that

$$w(x, y) = 1 \text{ if the site is the wall state;} \quad (9)$$

$$0, \text{ otherwise.}$$

In the hourglass simulation, an agent can be located only at a site where  $w(x, y) = 0$ . The angle of tangential direction is defined for each wall state site  $(x, y)$  and is represented by  $\theta_w(x, y)$ . The tendency of walking along a wall is defined by

$$\theta_{k,t} = R d\{\beta, \beta + \pi\} \quad (10)$$

$$\beta = R d\{\theta_w(x, y) | d(\mathbf{P}(k, t), (x, y)) \leq d(\mathbf{P}(k, t), (u, v)), w(x, y) = w(u, v) = 1, (x, y), (u, v) \in N_w\}, \quad (11)$$

where  $d((p, q), (x, y))$  represents the metric distance between two sites  $(p, q)$  and  $(x, y)$ , and  $N_w$  represents the neighborhood of wall-monitoring for each agent. If an agent is close to the wall with respect to  $N_w$ , the agent's velocity,  $\theta_{k,t}$  is parallel to

the tangential direction of the wall. After this operation, velocity matching (equation (7)) is applied to all agents. Only from (10) and (11) can agents close to the wall walk along the wall and other agents pass using the shortcut.

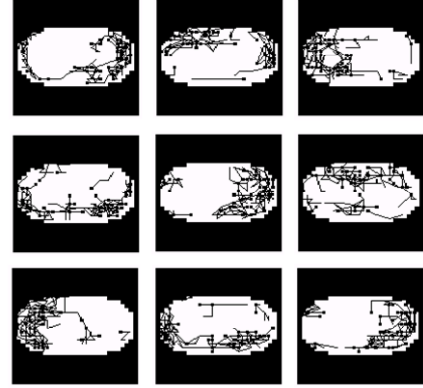


Figure 5 Snapshots of the time development of swarm move in the model simulation for the hourglass. Time proceeds from left to right and top to bottom. For this simulation,  $P = 10$ ,  $\alpha = 0.5$ ,  $L = 4$ , and  $r_a = r_f = 2$ . Each agent is represented by a black square with its own trajectory. First (top left) a main swarm is located at right hand, and then it moves to the left (top right). After that the swarm moves to the right again (middle center), and so on.

A solitary agent separated from a flockmate in our model undergoes a random walk because a potential transition is randomly chosen for each step. It follows that potential transitions stand for inherent noise. Whenever agents are highly concentrated, mutual anticipation can occur; inherent noise positively contributes to form a dense swarm. Thus, even if agents are exposed to large external perturbations, the perturbed transitions cannot be distinguished from inherent noise. A swarm resulting from mutual anticipation is thereby robust to external perturbation. In order to demonstrate the robustness of a swarm we implement “crab hour glass” (Nishiyama et al., 2011).

Fig. 5 shows snapshots of model simulations. It was assumed that an individual has a principal vector parallel to the tangent of the wall if it was close to the wall, in which the direction is chosen with uniform probability to be clockwise or anti-clockwise. Other rule settings were the same as in previous models. In the simulations, high concentrations initially occurred at the left or right ends, and the swarm rotated anti-clockwise. Most of the individuals walked along the wall, and some followed shortcuts. After a long period, the rotational direction reversed from clockwise to anti-clockwise and vice-versa.

The numbers of individuals in the divided areas (left, center, and right) shows regular oscillations (Fig. 6). The oscillating behavior of the model satisfies the properties of an hourglass of real soldier crabs. This oscillation mechanism is different from the periodic pattern of insect swarms based on escape-and-pursuit behavior.



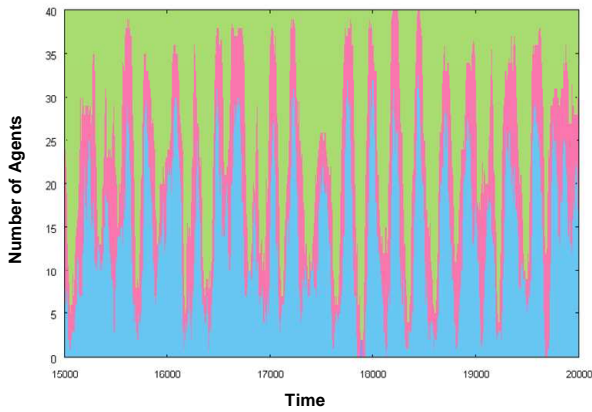


Figure 6 Numbers of agents in the left (blue), center (red) and right (green) areas of the container over time. For this simulation,  $P = 10$ ,  $\alpha = 0.5$ ,  $L = 4$ , and  $r_a = r_f = 2$ .

### Logic gates made of soldier crabs

A swarm is so robust that it can be used as a ball for collision-computing (Adamatzky, 2002). In addition to the hourglass model, we prepared a special scenario in which the area in which agents can move freely is tightly bounded by a wall, and there is a gradient of preferred direction. The constructed OR gate made of agents is shown in Fig. 7.

Each diagram of Fig. 7 shows a snapshot of the behavior of OR gate in time. Two entrances on the left represent input positions for two variables  $x$  and  $y$ , and one exit on the right represents the output position for  $x \text{ OR } y$ . If a swarm is present at position  $x$ , this state represents  $x=1$ . Agents move along the wall and rightward because of the gradient. After the collision of two swarms (each consists of 40 agents), the united swarm moves rightward and reaches the output position. It reveals  $x \text{ OR } y = 1$  for  $(x, y) = (1, 1)$ . Because  $x \text{ OR } y = 1$  for  $(x, y) = (0, 1)$  or  $(1, 0)$ , and  $x \text{ OR } y = 0$  for  $(x, y) = (0, 0)$ , this setup can implement the OR gate.

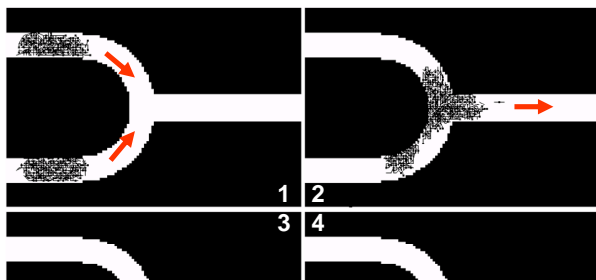


Figure 7 An OR gate of swarm balls. A swarm ball consists of 40 agents. Each agent is represented by a square with its 5-step trajectories. Four snapshots of a swarm at different time steps are

numbered. Red arrows represent the direction of motion of a swarm ball.

AND and NOT gates were also constructed using a swarm. Fig. 8 shows the behavior of an AND gate. In each diagram, two entrances on the left represent  $x$  and  $y$  for input, and the three exits on the right represent  $x \text{ AND } \text{NOT}(y)$ ,  $x \text{ AND } y$ , and  $\text{NOT}(x) \text{ AND } y$ , respectively. In the central exit on the right, there is a tidal pool in which a small swarm cannot enter. We define the tidal pool as a specific area  $U_p$  with the threshold  $c = 10$ . Because a swarm of 40 agents at the input position cannot enter the tidal pool, it retreats after the contact with the tidal pool and moves toward the output of  $x \text{ AND } \text{NOT}(y)$ .

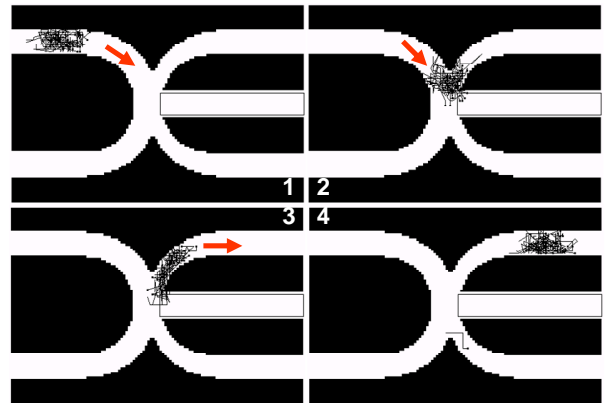


Figure 8 An AND gate of swarm balls. A swarm ball consists of 40 agents. Each agent is represented by a square with its 5-step trajectories. Four snapshots of a swarm at different time steps are numbered. Red arrows represent the direction of motion of a swarm ball.

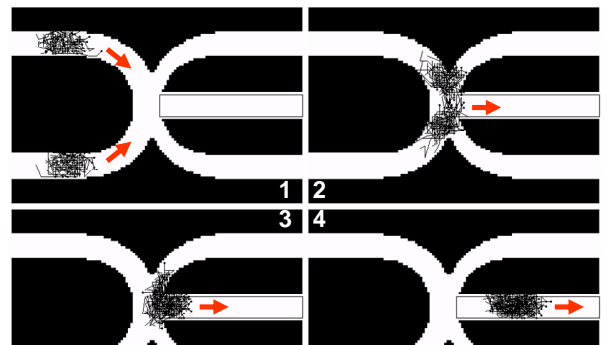


Figure 9 An AND gate of swarm balls. A swarm ball consists of 40 agents. Each agent is represented by a square with its 5-step trajectories. Four snapshots of a swarm at different time steps are numbered. Red arrows represent the direction of motion of a swarm ball.

Fig. 9 shows the behavior of an AND gate for  $(x, y) = (1, 1)$ . In this case, the collision of two swarms creates a large and united swarm, which enters the tidal pool. Thus, a united swarm produces the output of  $x$  AND  $y$ . If a swarm ball located at the input position of  $y$  is part of the logic gate,  $\text{NOT}(x)$  AND  $y$  can be utilized for  $\text{NOT}(x)$  for input  $x$ . Thus, this device can be utilized as a NOT gate. In this AND gate for  $(x, y) = (1, 1)$  sometimes the united swarm does not enter the water pool. It results in low performance (72%).

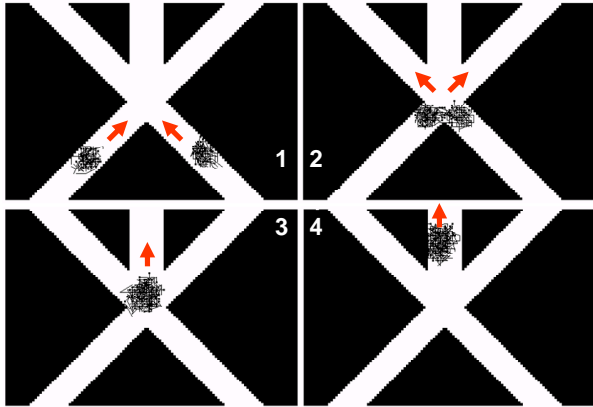


Figure 10 Another AND gate of swarm balls. A swarm ball consists of 40 agents. Each agent is represented by a square with its 5-step trajectories. Four snapshots of a swarm at different time steps are numbered. Red arrows represent the direction of motion of a swarm ball.

Thus we construct another AND gate as shown in Fig. 10. We set the initial principal directions of agents by the direction along the corridor represented by red arrows in the diagram 1 in Fig. 10. Actually we implement this gate by real soldier crabs. If the soldier crabs are set at the initial position and are threatened by a shadow suddenly appeared, they move straight. That is underlying implementation corresponding to the initial setting for principal vectors of agents. The swarms go straight. After the collision the united swarm moves following the united vectors. The performance of this AND gate is beyond 95%.

We here show three dynamics of our swarm model, water-crossing, hourglass and logic gate behaviors. The underlying mechanisms are based on mutual anticipation or inherent noise, which contribute to a robust, coherent swarm containing inherent turbulence. The characteristic flexibility and robustness of a swarm can be compared to a human's body awareness. A bird flock forms a large sub-domain that scales linearly with flock size. Because the proportion of the correlated domain against flock (body) size is constant, the flock appears to move as a single body (Cavagna et al., 2010). Our model can also show the scale-free correlation that has been observed in starlings and soldier crabs (Murakami et al., 2011). We believe that mutual anticipation is a key component in the generation and/or embedding of body awareness in a system. We now implement these logical gates by real soldier crabs, and the results will be given anywhere.

Robustness of a swarm plays an essential role in water-crossing, hourglass and logic gate behaviors. Because of robustness, a swarm can cross the water without being fallen into separated, hour glass shows periodic oscillation and logic gate shows high performance. In our model inherent noise (i.e. a number of potential transitions for each agent) contributes to make a robust swarm. Even if external perturbation is very large, the inherent noise cannot be distinguished from external perturbation. It entails that even external noise can coordinate to a robust swarm.

Even if the external noise increases, density and polarization of a swarm is not changed at all and a robust swarm is maintained, as shown in Fig. 11. The external perturbation is given by the product of the strength of a noise,  $\lambda$  and random variable,  $\xi$ . The random variable,  $\xi$ , is randomly chosen from  $[0, 1]$  and is coupled with velocity matching. When the projected velocity of agent is expressed as  $(v_x, v_y)$ ,  $v_x + \lambda\xi$  and  $v_y + \lambda\xi$  are given for the unit vector. In this simulation  $P = 20$ ,  $\alpha = 0.5$ ,  $L = 4$ , and  $r_a = r_f = 2$ . We tried other conditions with respect to  $P$ , and obtain similar results of polarization and density for  $10 < P < 20$ .

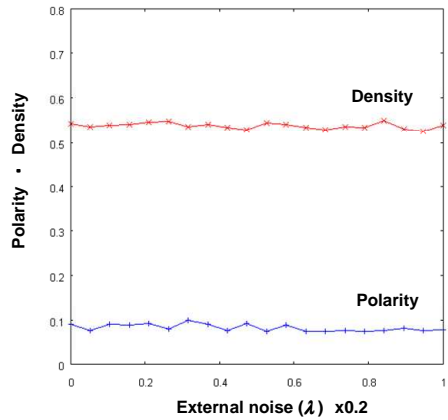


Figure 11. Polarization and density of a swarm generated by the model plotted against external noise.

In the next section, we discuss the significance of mutual anticipation to embodiment.

### Future and Past coordinate Present

The question whether a swarm, flock or school has a single consciousness or body has been addressed by investigating kinetic data from real animal groups and model simulations. A notable finding is that a swarm or a flock has a scale-free proportion of correlated domains, which reveals embodied collective behavior. Although it has been suggested that the interplay between anticipated states and memory states can contribute to a scale-free correlation in an asynchronous updating model (Gunji et al., 2011), our model is the first attempt to implement the interplay of anticipated and memory states in a swarm. We here discuss the relationships among the concept of body awareness, the interplay of anticipated and memory states, and flexibility and robustness.

Body awareness is studied in terms of a sense of ownership (SoO) and of a sense of agency (SoA) in neuroscience. It is known that SoO and/or SoA can be easily implanted in an object instead of a participant's own body through a synchronous interplay of visual and tactile stimuli (Ramachandran et al., 1996, 1998).

The generation of SoO and SoA in cognitive systems has also been studied. Fig. 10a shows a schematic diagram of SoO and SoA in sensory-motor coupling (Pfeifer, et al., 2007, Gallagher, 2000, Synofzik, 2008). After receiving a stimulus from the environment, a controller (brain) computes the anticipated state of its motor to adapt itself to the environment. The order from the controller is sent to the motor, and the actual state is revealed. A reaction from the environment is received again. In this scheme, the anticipated state processed on motor command is compared to the original intention in a controller. Because the comparison between the anticipated state and the original intention is executed before the motor moves, it constitutes a feed-forward process. In contrast, the actual state of the motor is compared to the original intention after the movement of the motor. This dynamic constitutes a feed-back process. SoA is thought to be related to a feed-forward and SoO to a feed-back process (Gallagher, 2000).

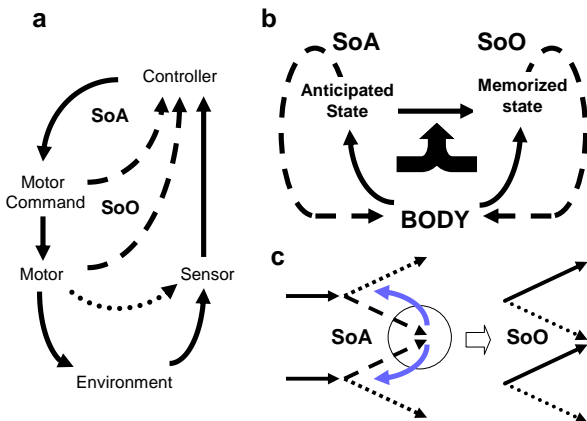


Figure 12. Sense of ownership (SoO) and sense of agency (SoA) in embodiment compared with body awareness in a swarm model. (a) Schematic diagram of SoO and SoA in a sensory-motor coupling system. (b) The schematic diagram of SoO and SoA in a system in which the sensory-motor distinction is vague in the “embodied body”. (c) SoA and SoO in our swarm model. Blue arrows represent back propagation in time from the anticipated popularity of transitions.

Because adaptive cooperation in a system entails the exploitation of decentralization and embodiment, the body includes redundancy resulting from reciprocal conflicts among components (Cruse et al., 2006). Such redundancy makes it possible to achieve complementary interplay between different modalities without forced learning and can result in a vague distinction between body and environment (Lungarella et al., 2006). The body and/or the boundary between the body and the environment is perpetually generated and maintained. The scheme involving SoO and SoA when the environment,

sensor and controller are mixed up in the form of body is represented in Fig. 12b. The motor command and motor are here represented by their featured, anticipated and memory states. In Fig. 12a, SoO and SoA constitute a hierarchical system. However, if there many redundant paths from controller to motor and embodiment between parts of the system can occur (i.e. the boundary of subsystems in a sensory-motor system becomes indefinite), the relationship between SoO and SoA in Fig. 12a can be replaced by that in Fig. 12b in which SoO and SoA are distributed in a parallel manner (Gunji, Sonoda & Niizato, 2011). The connection between SoO and SoA is dynamically generated to ensure consistency.

The dynamical connection between SoA and SoO is embedded in our swarm model. Through mutual anticipation, the anticipated popular sites are propagated backward in time, which can reveal actual transitions by asynchronous updating. Due to asynchronous updating and the avoidance of collisions by agents, a swarm is perpetually generated as a coherent system. These features can give rise to dynamic, flexible and robust swarming. After that, actual transition is memorized and is utilized as a principal vector to generate inherent noise (i.e., potential transitions) along the principal vector. SoO is here implemented as a reservoir to generate inherent noise or potentiality.

The underlying mechanisms of SoO and SoA emerge clearly in our swarm model. The interplay of anticipation and memory plays a central role in flexible and robust swarming. This interplay is characteristic of body awareness. Because a swarm is generated as a “body”, it can show a coherent density containing inherent turbulence. The swarm is robust to perturbed environments. A system that appears to be in equilibrium (e.g., a swarm ball or hourglass showing periodic oscillations) is in fact perpetually and robustly generated far from equilibrium. The idea of a “body” is thus well-defined not in the context of stability but of robustness.

## Conclusion

Based on the field observations of soldier crabs, *Mictyris guinotae*, we find that inherent noise can contribute to a dynamic and coherent swarm in which internal turbulence continuously flows. We implement such a phenomenon by an aggregation of agents of which each one have multiple potential transitions and can anticipate with each other. As a result we obtain dynamic and robust swarm even against the external perturbation.

Due to the robustness the swarm cannot be disturbed in perturbed environments such as water pool, and can be utilized as hour glass and logic gate. They are preliminary implemented by real soldier crabs, and that can be approximated by our model.

Since the swarm model is equipped with anticipation and memory, the model can be compared to the comparator model for SoA and SoO in body image, as long as a hierarchical structure is given up. Two loops including anticipated state or past state (memory) can cooperated with generating the current state. This structure is an essential structure to generate human body image. Our argument entails that a

swarm structure in our model can have a similar structure. Actually in the swarm a part of the swarm can be moved and operated by the swarm itself (corresponding to SoA). It results in a coherent and robust swarm as a whole not to be fallen into collapse of the swarm. This is the first step to connect the swarm with body image with respect to inherent time structure.

## References

- Adamatzky, A. Collision-based computing, Springer, 2002.
- Ballerini, M., Cabibbo, V., Candelier, R., Cisbani, E., Giardina, I., Lecomte, V., Orlandi, A., Parisi, G.P.A., Viale, M. and Zdravkovic, V. (2008a) Empirical investigation of starling flocks: A benchmark study in collective animal behavior. *Animal Behavior* 76: 201–215.
- Ballerini, M., Cabibbo, N., Candelier, R., Cavagna, A., Cisbani, E., Giardina, I., Lecomte, V., Orlandi, A., Parisi, G., Procaccini, A., Viale, M., and Zdravkovic, V. (2008b) Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *PNAS* 105: 1232-1237.
- Botvinick, M. and Cohen, J. (1998) Rubber hands ‘feel’ touch that eyes see. *Nature* 391: 756.
- Bradshaw, C. and Scoffin, T.P. (1999) Factors limiting distribution and activity patterns of the soldier crab *Dotilla myctiroides* in Phuket, South Thailand, *Mar Biol* 135: 83-87.
- Carere, C., Montanino, S., Moreschini, F., Zoratto, F., Chiarotti, F., Santucci, D., & Alleva, E. (2009) Aerial flocking patterns of wintering starlings, *Sturnus vulgaris*, under different predation risk. *Animal Behaviour* 77: 101-107.
- Cavagna, A., Cimarelli, C., Giardina, I., Parisi, G.S.R., Stefanini, F. and Viale, M. (2010) Scale-free correlation in the bird flocks. *PNAS*, 107: 11865-11870.
- Couzin, I.D. (2007) Collective minds. *Nature* 445: 715
- Couzin, I.D.(2008) Collective cognition in animal groups. *Trends in Cog.Sc.*13: 36-43.
- Cruse, H., Duerr, V. and Schmitz, J. (2007) Insect walking is based on a decentralised architecture revealing a simple and robust controller. *Philos. Trans. R. Soc. London A* 365: 221-250.
- Czir’ok, A., E. Ben-Jacob, I. Cohen, and T. Vicsek (1996) Formation of complex bacterial colonies via self-generated vortices. *Phyl Rev E* 54 (2): 1791-1801.
- Ehrsson, H.H., The experimental induction of out-of-body experience. *Science* 317, 1048 (2007).
- Gallagher, S. Philosophical conceptions of the self: Implications for cognitive science. *Trends Cognit. Sci.* 4, 14–21(2000).
- Gunji, Y-P., Niizato, T., Murakami, H., and Tani, I., (2010) Typ-Ken (an amalgam of type and token) drives Infosphere, *Knowledge, Technology and Policy* 23: 227-251.
- Gunji, Y.-P., Shirakawa, T., Niizato, T., Yamachiyo, M. and Tani, I. (2011) An adaptive and robust biological network based on the vacant-particle transportation model. *J. theor. Biol.* 272: 187-200.
- Gunji, Y-P., Sonoda, K. and Niizato, T. (2011) Embodiment and the paradox of the heap. *Radical Constructivism* (in press).
- Lenggenhager, B., Tadi, T., Metzinger, T. & Blanke, O., Video ergo sum: manipulating bodily self-consciousness. *Science* 317, 1096-1099 (2007).
- Lungarella, L., Sporns, O., Mapping information flow in sensorimotor networks. *PLoS Comp. Biol.* 2, 1301-1312. (2006).
- Murakami, H., Niizato, and Gunji, Y-P., Scale-free proportion by the model based on potential resonance. *PLoS Comp Biol* (submitted).
- Nishiyama, Y. Enomoto, K., Toda, M., Gunji, Y.-P., Hour glass behavior of soldier crab, *Mictyris guinotae*. 2011
- Peter J. F., Hsi-Te Shih, D. and Chan, B.K.K., (2010) A new species of *Mictyris guinotae* (Decapods, Brachyura, Mictiridae) from Ryukyu Island, Japan. *Crustaceana Monographs* 11: 83-105.
- Pheifer, R. and Scheier, C., Understanding Intelligence. The MIT Press, Cambridge, 2001.
- Pfeifer, R., Lungarella, M. and Iida, F., 2007, Self-organization, embodiment, and basically inspired robotics. *Science* 318, 1088-1093.
- Ramachandran, V.S., & Rogers-Ramachandran, D. Synaesthesia in phantom limbs induced with mirrors. *Phil. Trans. of the R. Soc. London. B, Biol. Sci.*, 263(1369), 377–386 (1996).
- Ramachandran, V. S., & Hirstein, W. The perception of phantom limbs. The D.O. Hebb lecture. *Brain*, 121, 1603–1630(1998)
- Reynolds, C.W. (1987) Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics*. 21(4): 25-34.
- Shih, J.T., (1995) Population-densities and annual activities of *Mictyris brevidactylus* (Stimpson, 1858) in the Tanshui mangrove swamp of northern Taiwan, *Zool. Stud.* 34: 96-105.
- Sumpter, D.J.T., *Collective Animal Behavior*, Princeton, Univ. Press, Princeton, 2010.
- Synofzik, M., Vosgerau, G. & Newen, G. Beyond the comparator model: A multifactorial two-step account of agency. *Consciousness and Cognition* 17, 219–239 (2008)
- Tsakiris, M., Prabhu, G. & Haggard, P. Having a body versus moving your body: how agency structures body-ownership. *Consciousness and Cognition* 15, 423-432 (2006).
- Varela, F.J., Evan, T. and Rosch, E., The Embodied Mind: Cognitive Science and Human Experience. Cambridge, MA: The MIT Press, 1992.
- Vicsek, Y., Czirok, A., Ben-Jacob, E. and Shochet, .(1995) Novel type of phase transition in a system of self-driven particles. *Phys Rev Let* 75: 1226-1229.
- Vicsek, T., Fluctuations and Scaling in Biology. Oxford Univ. Press, Oxford, 2001.
- Vicsek, T. and Zafiris A., Collective motion. *Arxiv preprint arXiv:1010.5017, 2010 - arxiv.org*.
- Yates, C. A., R. Erban, C. Escudero, I. D. Couzin, J. Buhl, I. G. Kevrekidis, P. K. Maini, and D. J. T. Sumpterh (2009) Inherent noise can facilitate coherence in collective swarm motion, *PNAS* 106: 5464-5469.

# Explaining Emergent Behavior in a Swarm System Based on an Inversion of the Fluctuation Theorem

Heiko Hamann<sup>1</sup>, Thomas Schmickl<sup>1</sup>, Karl Crailsheim<sup>1</sup>

<sup>1</sup>Artificial Life Laboratory of the Department of Zoology, Karl-Franzens University Graz, Universitätsplatz 2, A-8010 Graz, Austria  
heiko.hamann@uni-graz.at

## Abstract

A grand challenge in the field of artificial life is to find a general theory of emergent self-organizing systems. In this paper we try to explain the emergent behavior of a simulated swarm by applying methods based on the fluctuation theorem. Empirical results indicate that the swarm is able to produce negative entropy within an isolated sub-system due to ‘frozen accidents’. Individuals of the swarm are able to locally detect fluctuations of the global entropy measure and store them, if they are negative entropy productions. By accumulating these stored fluctuations over time the swarm as a whole is producing negative entropy and the system ends up in an ordered state. We claim that this indicates the existence of an inverted fluctuation theorem for emergent self-organizing dissipative systems. This approach bears the potential of general applicability.

## Introduction

One characteristic of living organisms is their metabolism. Living beings require energy in order to maintain their internal order. This is determined by the second law of thermodynamics that describes the ubiquitous decay of all things and does not allow the increase of order without the cost of dissipation. In the context of self-organizing systems one might cite Parunak and Brueckner (2001): “Emergent self-organization in multi-agent systems appears to contradict the second law of thermodynamics.” This is of course not the case, as discussed by Parunak and Brueckner (2001), one has to distinguish between two kinds of sub-systems: one that hosts the self-organizing swarm and one in which disorder is increased. Hence, a swarm can be thought of as a heat pump that decreases entropy<sup>1</sup> in one basin in favor of increased entropy in another basin. However, the question of how the swarm manages to do that still persists. Whether thermodynamic properties are relevant and helpful in understanding such systems is currently discussed (Polani, 2008; Hamann et al., 2011a).

<sup>1</sup>In principle, we refer here to Gibbs entropy  $S = -k_B \sum_i p_i \ln p_i$ , for Boltzmann constant  $k_B$  and the sum over all microstates with probabilities  $p_i$  which applies especially to classical, finite systems far away from equilibrium. However, an intuitive understanding of entropy suffices in the following.

The emergence of life is explained by natural selection in combination with random events (natural evolution). It is one thing to select the adapted organism but the mutation, that results in an improved adaptivity, has to occur first. Concerning the genetic code Crick (1968) phrased the term ‘Frozen Accident Theory’. While Crick was introducing this concept with focus on genetics, Gell-Mann (1995) applied it to everything:

[...] the effective complexity [of the universe] receives only a small contribution from the fundamental laws. The rest comes from the numerous regularities resulting from ‘frozen accidents’.

The intuition of this theory is relatively clear in the context of the slow evolution of our universe. However, we want to define a concept of frozen accidents within emergent self-organizing multi-agent systems (De Wolf and Holvoet, 2005) that explains how they can work as heat pumps in the sense as described above.

While a heat pump has to work against the second law (e.g., diffusion of heat) by expending energy, limited violations of the second law without the expenditure of energy (Evans et al., 1993) are also possible as, for example, indicated by Maxwell (1878):

The truth of the second law is ... a statistical, not a mathematical, ... for it depends on the fact that the bodies we deal with consist of millions of molecules.

Violations of the second law are possible for small systems and short time scales, that is, at atomic and micron scales over short times (up to two seconds), and were shown experimentally (Wang et al., 2002). We claim that the reduction of entropy by emergent self-organizing systems could be explained by the ‘summation’ of such violations to the second law. The second law is only statistical and, hence, allows spontaneous decreases of entropy in isolated systems with nonzero probability.

The possibility of temporal entropy decreases exists because a system at a temperature above absolute zero according to statistical mechanics always shows thermal fluctuations, that



are random deviations of a system from its equilibrium. Say  $x$  is a thermodynamic variable (i.e., it describes a state of a thermodynamic system at a given time) then the probability distribution  $f(x)$  of this variable for a system at maximum entropy (at equilibrium state) turns out to be Gaussian with mean  $\mu = 0$ :

$$f(x) = \frac{1}{\sqrt{2\pi\langle x^2 \rangle}} \exp\left(-\frac{x^2}{2\langle x^2 \rangle}\right), \quad (1)$$

for the variance defined by the mean square fluctuation  $\sigma^2 = \langle x^2 \rangle$ , which is an average over many ensembles (i.e., average over many realizations of the system). Hence, the probability of observing negative ( $\int_{-\infty}^0 f(x)dx$ ) or positive fluctuations ( $\int_0^{+\infty} f(x)dx$ ) is equal at equilibrium.

The fluctuation theorem (Evans and Searles, 2002; Evans et al., 1993) quantifies the probability of violations to the second law. For short intervals it can be said that nature was running in reverse. Even concerning living systems this might be true. For example, small ‘machines’ within a cell (e.g., mitochondria) are likely to run in reverse from time to time. A transfer of this concept to the macro-world is typically denied categorically. In a review of Wang et al. (2002), Gerstner (2002) wrote: “For larger systems over normal periods of time, however, the second law of thermodynamics is absolutely rock solid.”

Generally the fluctuation theorem is said to be applicable only to the micro-world, where Brownian motion can be observed. Truly, this is a well chosen hypothesis. However, what if we allow dissipation of energy in the first place, separate the system in two sub-systems of the self-organizing part and a heat bath, and then observe only the behavior in the self-organizing half of the system? That way one could argue that we simulate the micro-world by a macro-system at the cost of lost heat. This concept (see Fig. 1) is for example taken into account by Smith (2008) when stating

$$dQ = -k_B T dS \equiv k_B T d\mathcal{I}, \quad (2)$$

for an increment of heat  $dQ$  rejected by the system to a thermal bath at temperature  $T$ , Boltzmann constant  $k_B$ , reduction in entropy of the (sub-)system’s internal state  $-dS$ , and the increase in information  $d\mathcal{I}$  (note that Smith (2008) defines information as “the reduction in some measure of entropy”). Note that the mere property of being dissipative is not sufficient to explain a self-organizing system. In addition to squandering energy the system has to generate orderly structures. Dissipation is only a necessary condition for negative entropy production while additional sufficient conditions exist. In case of Rayleigh-Bénard convection (Bodenschatz et al., 2000), for example, initially fluctuating flows (Wu et al., 1995) occur that are enhanced and trigger the formation of Bénard cells in spontaneous symmetry breaking, cf. also (Nicolis and Prigogine, 1977; Haken,

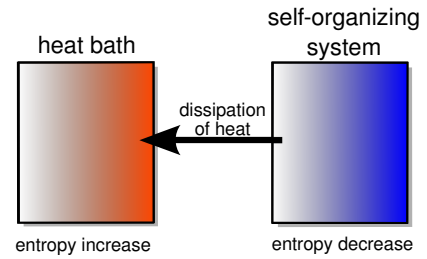


Figure 1: Schematic of a system divided into a heat bath with increasing entropy and a self-organizing, dissipative sub-system with decreasing entropy.

1977). We want to point out the self-amplification of fluctuations as such a sufficient condition here.

In this paper, we report empirical evidence that the negative entropy production in emergent self-organizing systems is based initially on frozen accidents allowed by the original fluctuation theorem which, in turn, leads in the end to a global behavior that is described by an inversion of the fluctuation theorem in dissipative self-organizing systems. This concept might bear potential of embedding the concept of emergent behavior in multi-agent systems (swarms, self-propelled particles etc.) in a theoretical framework built on sound foundations of theories from physics. Hence, we propose an approach to understand emergent behavior through thermodynamics which follows up our earlier reported concept (Hamann et al., 2011a).

In addition, the relation to the fluctuation theorem might allow to define preconditions for effective self-organizing systems in the future. For example, one can define minimum requirements for the agents of the system concerning its cognition abilities in order to be able to leverage fluctuations. The agent needs sensors that allow to estimate at least probabilistically whether the (local) entropy has just decreased. Furthermore, the system needs the ability to store such local fluctuations.

In the following we describe the investigated scenario and the fluctuation theorem. We analyze the multi-agent system or swarm, discuss how the results could be viewed as obeying an inverted fluctuation theorem and conclude by giving a short summary and outlook.

## BEECLUST algorithm

The BEECLUST algorithm can be considered a model algorithm for swarms. It is based on observations of young honeybees (Szopek et al., 2008), was analyzed in many models (Hereford, 2011; Schmickl and Hamann, 2011; Schmickl et al., 2009; Hamann et al., 2011b, 2010), and even implemented in a swarm of robots (Schmickl et al., 2008).

This algorithm allows a swarm to aggregate at a maximum of a gradient field although individual agents do not perform a greedy gradient ascent. Hence, it might be justified to

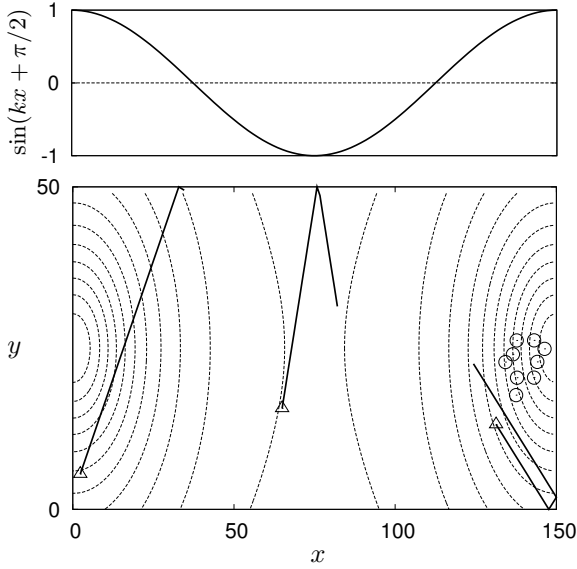


Figure 2: Bottom: Typical state of a swarm controlled by BEECLUST; positions of stopped agents (circles) and moving agents (triangles) with trajectories of the last 20 time steps, contours show levels of the gradient field. Top: function used in eq. 7.

- 1.) Each agent moves straight until it perceives an obstacle  $O$  within sensor range.
- 2.) If  $O$  is a wall the agent turns away and continues with step 1.
- 3.) If  $O$  is another agent and there is a third agent as well, the agent measures the local gradient value. The higher the gradient value the longer the agent stays still. After this waiting period, the agent turns away from the other agent and continues with step 1.

Figure 3: The BEECLUST algorithm (stop threshold of 3).

call this emergent behavior. Controlled by this algorithm three agents will stop (note that in previous works typically a threshold of two was chosen, which is, however, irrelevant in this paper) when they approach each other, measure the local value of the gradient, and wait for some time proportional to this measurement. Clusters form and finally the swarm will be aggregated close to the global optimum of the gradient field (see the lower part of Fig. 2). See Fig. 3 for a definition of the BEECLUST algorithm.

The collective aggregation close to the global optimum is achieved via a positive feedback process (Hamann et al., 2011b): Clusters of 3 stopped agents will form by chance anywhere in the arena. Agents in clusters closer to the global optimum have longer waiting times. These clusters will exist longer than those that are farther away from the global opti-

Table 1: Used parameter setting in this work.

arena dimensions	$150 \times 50$ [length units] <sup>2</sup>
proximity sensor range	3.5[length units]
max. waiting time	660[time units]
velocity	3[length units]/[time units]
number of agents	25

mum. Hence, the chance of growing into a cluster of size 4 is bigger for clusters closer to the global optimum. The area covered by clusters grows with the number of contained agents and clusters covering a bigger area are more likely to be approached by chance by moving agents. Hence, bigger clusters will grow faster. This process, typically, leads to just one big cluster close to the global optimum. The agents interact only locally and a BEECLUST-controlled swarm is able to break symmetries (Hamann et al., 2011b). Hence, this behavior is different from other aggregation processes, for example, star formation which includes global interactions due to gravitation.

In the following experiments, the agents have initially random headings, are in the state ‘moving’, and are random uniformly distributed in the arena. The gradient field is bimodal with maxima of the same value and shape (see contours in Fig. 2). See Table 1 for the standard parameters used.

## Fluctuation Theorem

According to Evans and Searles (2002) the group of fluctuation theorems “gives an analytical expression for the probability of observing Second Law violating dynamical fluctuations in thermostatted dissipative non-equilibrium systems.” In a thermostatted system the temperature is kept constant, for example, by rescaling the particles’ velocities. The system can be thought of as being in contact with a large heat reservoir in order to thermostat the system. One of these theorems (steady state fluctuation theorems) applies to time-reversible, thermostatted, ergodic dynamical systems and defines the relation of fluctuations (Evans and Searles, 2002)

$$\lim_{t \rightarrow \infty} \frac{1}{t} \ln \frac{P[\bar{\Sigma}_t = A]}{P[\bar{\Sigma}_t = -A]} = A, \quad (3)$$

for the time averaged entropy production  $\bar{\Sigma}_t = (1/t) \int_0^t \Sigma(s) ds$ . The fluctuation theorem compares probabilities of observing a certain time averaged entropy production  $A$  to its negative value  $-A$ . The value  $P(\bar{\Sigma}_t = A)$  describes the probability of finding the system initially in those states that subsequently generate bundles of trajectory segments with the time averaged value  $A$ . The above theorem (eq. 3) predicts an exponential increase of the relation  $P(\bar{\Sigma}_t = A)/P(\bar{\Sigma}_t = -A)$ . Hence, with increasing time positive entropy producing trajectories become exponentially more likely than their negative entropy producing counterparts.



As a consequence of the fluctuation theorem one obtains the Second Law Inequality

$$\langle \bar{\Sigma}_t \rangle \geq 0, \quad \forall t, \quad (4)$$

which states that the average over many ensembles, in which the time averaged entropy productions were measured, is positive. Hence, the fluctuation theorem is in accordance with the second law of thermodynamics.

### Analysis of BEECLUST

We consider a system of  $N$  agents that move in a two-dimensional box and gradient field. We assume the particles to move frictionless which basically means they have a permanent acceleration compensating friction. This, in turn, means they have an energy reservoir (cf. active particles (Schweitzer, 2003)) and permanently dissipate heat which results in a situation as shown in Fig. 1. In addition, we allow infinite accelerations because the agents stop and start within one time step in our numerical simulation. Energy costs have to be paid to allow self-organization and to comply with the second law of thermodynamics. In the following we carry out the separation between these two sub-systems: the self-organizing sub-system containing the agents and the sub-system typified by the heat reservoir. Due to its energy dissipation the self-organizing sub-system does not have to obey the second law of thermodynamics. We define the following equations of motion for each agent  $i$

$$\dot{\mathbf{q}}_i = \mathbf{p}_i/m, \quad (5)$$

$$\dot{\mathbf{p}}_i = \mathbf{F}_i + \begin{cases} -\mathbf{p}_i, & \text{particle autonomously stops} \\ \mathbf{p}'_i, & \text{particle autonomously starts} \\ 0, & \text{else} \end{cases}, \quad (6)$$

where  $\mathbf{q}_i = (x_i, y_i)^\top$  is the position of agent  $i$ ,  $\mathbf{p}_i$  is the momentum, and  $\mathbf{p}'_i$  is the value of  $\mathbf{p}_i$  at the time the agent had stopped. We have  $\mathbf{F}_i > 0$  in case the agent bounces off the bounds or closely approaches another agent. This can be implemented, for example, via a WCA potential (Weeks et al., 1971), which is a purely repulsive potential. As thermostat method we use velocity scaling which is governed by the number of stopped agents. In particular, the special periods of time in which all agents are stopped are converted to time periods of no extend. Note that this is only our method of measuring the self-organizing system. It is not intrinsic to the system and the behavior of the agents is unconcerned by this method.

The system dynamics takes place in a high dimensional phase space  $(\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_{N-1}, \mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{N-1}) \in \Gamma$ . In the following we need to detect the essentials of this dynamics by a measure of entropy. We ignore the momenta  $\mathbf{p}$  and also the y-positions because the main feature of the clusters is defined by the agents' x-positions (see Fig. 2). Ignoring

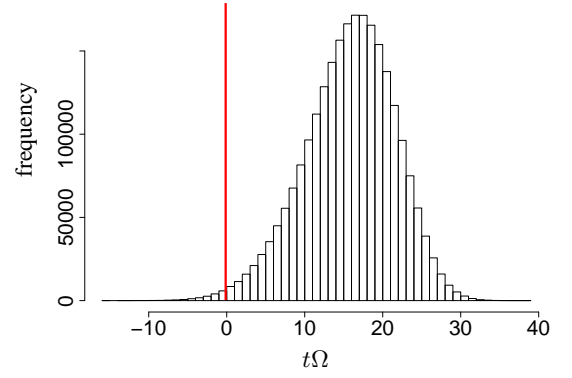


Figure 4: Distribution of the entropy production for a swarm controlled by the BEECLUST algorithm,  $t = 1500$ ,  $\langle t\Omega \rangle \approx 15.77$ ,  $T = 909.1$ , number of samples  $n \approx 5.0 \times 10^6$ .

the momenta does not hide entropy. Although we start with all nonzero momenta and during the experiments we have inhomogeneous momentum distributions but the experiments typically end with almost all agents stopped (i.e., again a homogeneous momentum distribution). Similar to (Evans and Searles, 2002, Sec. 4.3) we observe the agent density modulation via

$$\rho(k, t) = \sum_{i=1}^N \sin(kx_i(t) + \frac{\pi}{2}), \quad (7)$$

where  $x_i(t)$  is the x-position of agent  $i$  at time  $t$ ,  $k = 2\pi/L$ , and  $L = 150$  is the box length. The applied sine-function is shown in Fig. 2. Agents in the leftmost and rightmost quarters of the arena contribute positively, agents in the middle contribute negatively. In equilibrium,  $x_i \in [0, L]$  would be equally distributed averaged over many ensembles, yielding  $\langle \rho \rangle = 0$ . By applying the converse argument, averages of  $\langle \rho \rangle \neq 0$  would correspond to unequal distributions of agents whereas negative and positive values indicate whether the main cluster is in the middle or at the ends.

Following Evans and Searles (2002) we define a ‘dissipation function’  $\Omega(\Gamma)$  that gives the entropy production for a given phase space trajectory. We integrate changes of  $\rho$  over a time interval  $[0, t]$

$$t\Omega = \beta \int_0^t \dot{\rho}(k, s) ds = \beta(\rho(k, t) - \rho(k, 0)) \quad (8)$$

and

$$\beta = 1/T = \left( \frac{2}{k_B N_d} \left\langle \sum_{i \in [0, N-1]} \frac{\mathbf{p}_i^2}{2m} \right\rangle \right)^{-1} \quad (9)$$

is the reciprocal temperature of the initial ensemble with Boltzmann constant  $k_B$  and degrees of freedom  $N_d = 2N$ . The distribution of the entropy production for  $N = 25$  agents controlled by the BEECLUST algorithm, which were

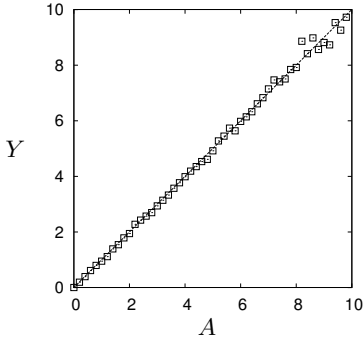


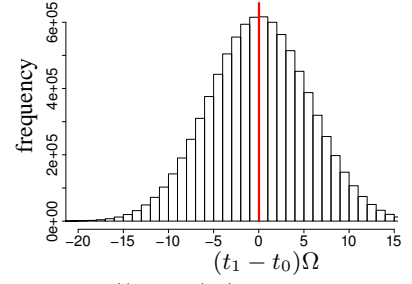
Figure 5: Test of the entropy production distribution of the BEECLUST-controlled swarm shown in Fig. 4 against the fluctuation theorem (eq. 10),  $Y = \frac{1}{\beta t} \ln \frac{P[\rho(k,t) - \rho(k,0) = A]}{P[\rho(k,t) - \rho(k,0) = -A]}$ ,  $t = 1500$ ,  $T = 909.1$ . Note that any  $Y \neq 0$  corresponds to negative entropy production.

initially random uniformly distributed, is shown in Fig. 4 for  $t = 1500$ . The initial uniform distribution yields  $\langle \rho(0) \rangle = 0$  which is the state of maximal entropy. Hence, any distribution of the entropy production with a mean of  $\langle t\Omega \rangle \neq 0$  indicates negative entropy production (i.e., averaged differences of the density modulation can have negative or positive signs but imply negative entropy production, if they are nonzero). The ensemble average is  $\langle t\Omega \rangle \approx 15.77$  which means that negative entropy is produced (initially at maximum entropy). Note that there is no direct influence by the gradient field to the entropy productions which are defined based on the agents' x-positions. Furthermore, the waiting times, that are determined by the gradient field, vary only by a factor of 5 between the minimum and the maximum. Now we want to apply the fluctuation theorem (eq. 3) to this system. Especially we have to assume time-reversibility which is problematic because BEECLUST-controlled systems are in general not reversible (Hamann et al., 2011a). However, we argue that it is fair to assume approximate reversibility because the irreversibility vanishes, if the agents measure almost equal gradient values (typically the difference is only about  $\pm 10\%$ ) determining almost equal waiting times and almost equal wake-ups. Applying the fluctuation theorem gives

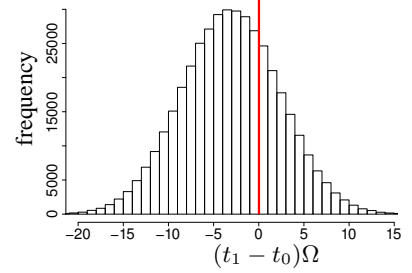
$$\lim_{t \rightarrow \infty} \frac{1}{t} \ln \frac{P[\rho(k,t) - \rho(k,0) = A]}{P[\rho(k,t) - \rho(k,0) = -A]} = \beta A. \quad (10)$$

The data shown in Fig. 4 is tested whether it obeys eq. 10 in Fig. 5. The fluctuation theorem is satisfied for this system although the system is producing negative entropy and actually abandoning the equilibrium to which it was initialized. Hence, one could speak of an 'inverted fluctuation theorem' that is satisfied here.

In the following we want to investigate how it is possible for this self-organizing system to produce negative entropy. We hypothesize that the negative entropy production is based on



(a) no stopping,  $\langle (t_1 - t_0)\Omega \rangle \approx 0.06$ ,  $n \approx 8.6 \times 10^6$



(b) stopping,  $\langle (t_1 - t_0)\Omega \rangle \approx -3.09$ ,  $n \approx 4.1 \times 10^5$

Figure 6: Distributions of the entropy production for an early time interval during the transient ( $t_0 = 15$ ,  $t_1 = 20$ ,  $T = 909.1$ ) classified according to whether a stopping agent was observed during the measurement.

fluctuations and the stopping behavior of the agents, hence, a process of frozen accidents. Note that such a mechanism is similar to the famous thought experiment 'Maxwell's Demon' (Maxwell, 1871). Furthermore, an implementation of Maxwell's Demon was reported (Bannerman et al., 2009) that is used as a cooling technique (cf. our metaphor of a heat pump in the introduction). Here we have rather a 'distributed demon' embodied by many autonomous agents that control themselves (Adami (1998) applies a similar argument to evolution). BEECLUST does not sort particles or agents as Maxwell's Demon but aggregate them (i.e., we generate uneven density distributions).

We measure the entropy production within a limited time interval  $[t_0 = 15, t_1 = 20]$  in the early transient. In addition, we classify for each measurement whether at least one agent changed its state from moving to stopped (starting agents do not occur that early in the simulation). The entropy production distribution for these two classes are shown in Fig. 6. For the measurements without a stopping agent the averaged change in the density modulation is about 0 ( $\langle (t_1 - t_0)\Omega \rangle \approx 0.06$ ). In contrast, for those measurements with stopping agents the averaged change of density modulation is negative ( $\langle (t_1 - t_0)\Omega \rangle \approx -3.09$ ) indicating frozen accidents. For much later time intervals no difference between measurements with stopped and without a stopping agents are found. The negative value of  $\langle (t_1 - t_0)\Omega \rangle$  demands for clarification because in the limit  $t \rightarrow \infty$  the average density modulation is positive.

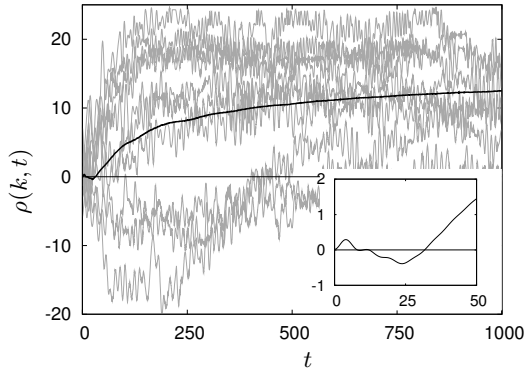


Figure 8: Evolution of the agent density modulation over time, black line shows ensemble average, gray lines show samples, insert shows details of the ensemble average within the first 250 time steps.

The explanation is a special feature of the BEECLUST-controlled swarm in this scenario which consists of three phases (see Fig. 7). In the short period before the first cluster forms, the average entropy production is  $\Omega = 0$  indicating that the original fluctuation theorem holds for this phase. The first cluster usually does not form close to the global optima but relatively close to the middle of the arena, see Fig. 7(a). In this area the agent density modulation (eq. 7) contributes negatively. In a second phase the average density modulation is negative ( $\Omega < 0$ ) because the density close to the middle of the arena increases further, see Fig. 7(b). This is also indicated by the evolution of the agent density modulation over time as shown in Fig. 8. Initially it stays close to 0 and only later it clearly takes a positive sign. The insert shows details of the first 250 time steps and indicates negative slope for the time interval  $[15, 20]$  (i.e., second phase) of Fig. 6. Only later the clusters ‘move’ towards the ends of the arena probably due to wall effects, see Fig. 7(c) and consequently the average density modulation is positive ( $\Omega > 0$ ).

### Discussion

Note again that  $\rho(k, t) = 0$  corresponds to maximum entropy. Therefore, any  $\rho(k, t) \neq 0$  in Fig. 8 indicates negative entropy production. We conclude that the negative entropy production of this system is initiated by entropy fluctuations which are normally distributed and are negative/positive with about the same probability respectively according to the original fluctuation theorem and as seen in Fig. 6(a). Some of these ‘negative entropy production’-events are locally observable by the agents themselves because there are three agent-to-agent encounters with mutual perception. This local perception of the global measure of entropy is leveraged by stopping all three agents and stores the local entropy fluctuation. Cascades of such stopping behaviors generate a positive feedback (self-amplification of fluctuations as in Rayleigh-Bénard convection). In the end,

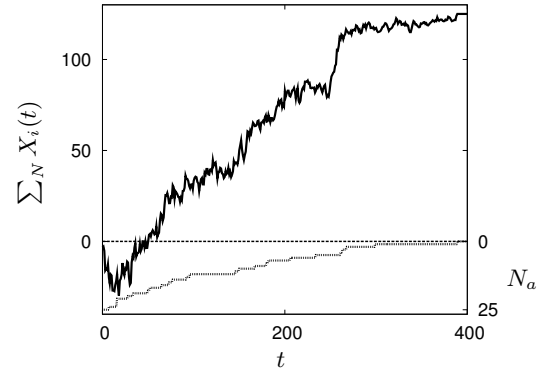


Figure 9: Sample run of a simple model based on summations of  $N = 25$  random processes initialized to  $X_i(0) = 0$  and based on normally distributed random variables ( $\mu = 0$ ,  $\sigma^2 = 1$ ).

a system dynamics is generated, that can be described by an inverted fluctuation theorem, which dictates an exponentially increasing probability of low entropy states. Hence, this emergent self-organizing swarm does indeed rely on frozen accidents. Note that the overall system still produces positive entropy (e.g., due to accelerations of the agents) while the agent-position-based entropy is only reduced in the self-organizing sub-system.

The effectiveness of the frozen-accidents concept can easily be made clear by a simple model. We represent the entropy contribution of each agent  $i$  by a random process  $X_i(t)$ . The total entropy is just the sum  $\sum_{i=0}^N X_i(t)$  over all agents  $N$ . The restriction of all random processes to the interval  $[-5, 5]$  is essential and we define  $X_i(t) = 5$ ,  $\forall t > t_0$  with  $t_0$  is the first time agent  $i$  achieved  $X_i(t_0) = 5$ . That is, once a random process reaches  $X_i(t_0) = 5$  (a local property) it stays there forever—a frozen accident. As a consequence the number of active random processes  $N_a$  will decrease monotonically. A sample run of this simple model for  $N = 25$  based on Gaussian distributed  $X_i$  and initialization  $X_i(0) = 0$  is shown in Fig. 9. The bias in the otherwise random trajectory is noticeable. Note that the summation of Gaussian distributed random variables  $\sum_N X_i$  with each having a variance of  $\sigma_i^2$  results in a random variable that is also Gaussian distributed with a variance of  $\sigma^2 = \sum_N \sigma_i^2$ . With decreasing number of active processes  $N_a$  more and more variances vanish ( $\sigma_i^2 = 0$ ). Hence, also the variance of the sum will decrease which is the macroscopic effect of the frozen accidents and ensures that states of low entropy are much more likely to be maintained.

The results shown in Figs. 4, 5, and 6(b) indicate that this emergent self-organized system obeys an inversion of the fluctuation theorem which could be stated as

$$\lim_{t \rightarrow \infty} \frac{1}{t} \ln \frac{P[\bar{\Sigma}_t = -A]}{P[\bar{\Sigma}_t = A]} = A, \quad (11)$$

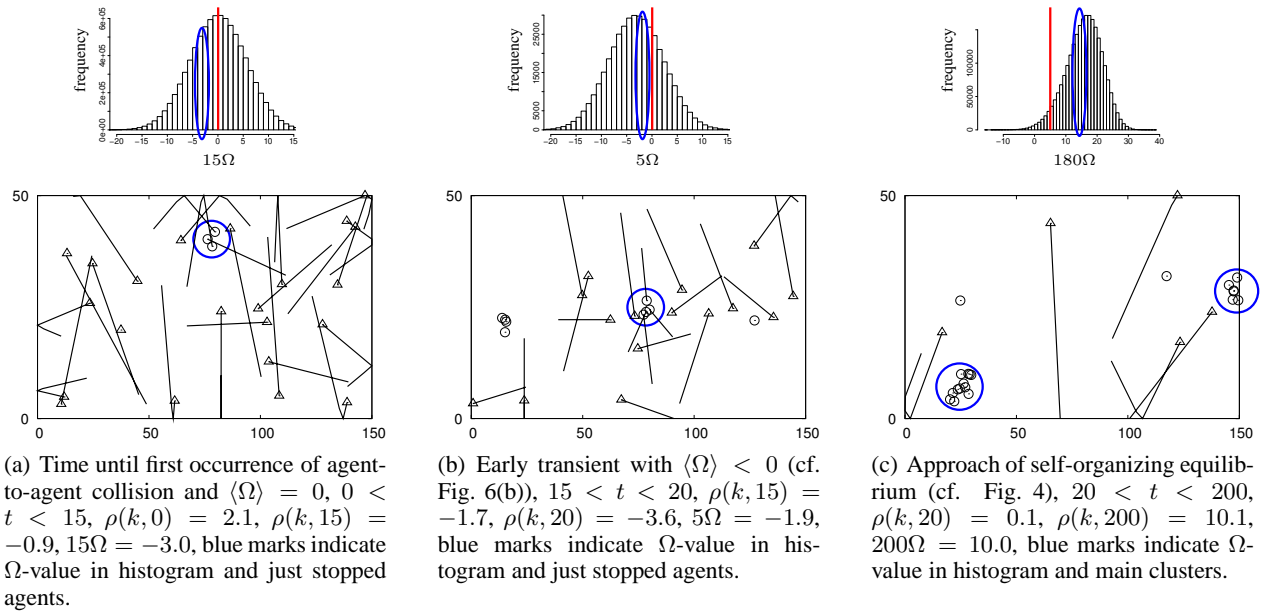


Figure 7: The three phases observed in the investigated scenario each with a representative entropy production histogram and a plot of the arena showing moving (triangles) and stopped agents (circles) with a line indicating their most recent trajectory (histograms are meant to be qualitative).

following eq. 3. We get an immediate interpretation of this self-organizing system by inverting the interpretation of the fluctuation theorem. A self-organizing system that is started with high entropy will produce negative entropy with an exponentially increasing probability over time. As a consequence there is a ‘self-organization equilibrium’ of lower entropy to which the system will converge. As a second consequence the self-organizing entropy-reduction behavior is a transient phenomenon, cf. (Prigogine, 1997, p. 62).

## Conclusion

In this paper, we have analyzed an emergent self-organizing multi-agent (or swarm) system controlled by the BEECLUST algorithm with methods based on and suggested by the fluctuation theorem. The results provide empirical evidence for the existence of an inverted fluctuation theorem that applies for such dissipative self-organizing systems. In addition, this work suggests the rich and thought-provoking metaphor of considering emergent swarm systems as implementations of a ‘distributed Maxwell’s demon’ because random events are leveraged by autonomous decisions of embodied agents based on locally measured samples of a global entropy change. A theory based on an inverted fluctuation theorem could prepare a wide basis for the analysis of self-organizing systems. We claim these methods have a potential for general applicability. For example, in flocking dissipation occurs due to rotational accelerations and averaging of directions (loss of information). Potential generalization is also indicated by preliminary results in

other scenarios which will be reported in future work.

Specific exemplary benefits of such a theory could be the definition of preconditions for self-organization, for example, concerning the cognitive abilities of the agents. Statistical properties of fluctuations describe the time-scales on which negative entropy production can be observed. The agents need to perceive local samples of this global property of negative entropy production and need to react within these time-scales. Hence, conditions for controller sampling rates could be derived. The agents need appropriate sensors that allow local measurements of entropy with an accuracy that is sufficiently higher than the rate at which events of negative entropy production occur. Thus, conditions for successfully generating positive feedbacks could be derived.

Especially the origin of BEECLUST confirms the possibility of applying the proposed methods to natural systems such as clustering behaviors in young honey bees (Szopek et al., 2008) or other social insects, as well as flocks, herds, and shoals. Hence, the same methods could be used for artificial and natural systems which could, in turn, enrich primarily biological studies.

This work proved again that thermodynamics offers many fully developed methods which can often be applied even unmodified to problems of emergent behavior (cf. Hamann et al. (2011a)). Pursuing this research track might be a promising way of achieving general insights to still rather fuzzy concepts such as emergence or self-organization.

Finally, it is clear that the reported approach is truly interdisciplinary in combining methods and problems from physics,

biology, and computer science. It is obvious that, at least in the field of artificial life, any future research success has to be founded on a collection of several scientific fields. In our future work, we hope to continue this approach by generalizing the concept of an inverted fluctuation theorem for emergent self-organizing multi-agent systems.

## Acknowledgments

Supported by: EU-IST-FET project ‘SYMBRION’, no. 216342; EU-ICT project ‘REPLICATOR’, no. 216240.

## References

- Adami, C. (1998). *Introduction to artificial life*. Springer-Verlag.
- Bannerman, S. T., Price, G. N., Viering, K., and Raizen, M. G. (2009). Single-photon cooling at the limit of trap dynamics: Maxwell’s demon near maximum efficiency. *New Journal of Physics*, 11(6):063044.
- Bodenschatz, E., Pesch, W., and Ahlers, G. (2000). Recent developments in Rayleigh-Bénard convection. *Annual review of fluid mechanics*, 32(1):709–778.
- Crick, F. H. (1968). The origin of the genetic code. *Journal of Molecular Biology*, 38(3):367–379.
- De Wolf, T. and Holvoet, T. (2005). Emergence versus self-organisation: Different concepts but promising when combined. In Brueckner, S., Serugendo, G. D. M., Karageorgos, A., and Nagpal, R., editors, *Proceedings of the workshop on Engineering Self Organising Applications*, volume 3464 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag.
- Evans, D. J., Cohen, E. G. D., and Morriss, G. P. (1993). Probability of second law violations in shearing steady states. *Physical Review Letters*, 71:2401–2404.
- Evans, D. J. and Searles, D. J. (2002). The fluctuation theorem. *Advances in Physics*, 51(7):1529–1585.
- Gell-Mann, M. (1995). Plectics. In Brockman, J., editor, *The Third Culture: Beyond the Scientific Revolution*, pages 316–332. Touchstone Press, New York.
- Gerstner, E. (2002). Second law broken: Small-scale energy fluctuations could limit minaturization. *Nature online news*.
- Haken, H. (1977). *Synergetics - an introduction*. Springer-Verlag, Berlin, Germany.
- Hamann, H., Meyer, B., Schmickl, T., and Crailsheim, K. (2010). A model of symmetry breaking in collective decision-making. In Doncieux, S., Girard, B., Guillot, A., Hallam, J., Meyer, J., and Mouret, J., editors, *From Animals to Animats II*, volume 6226 of *LNAI*, pages 639–648. Springer-Verlag.
- Hamann, H., Schmickl, T., and Crailsheim, K. (2011a). Thermodynamics of emergence: Langton’s ant meets Boltzmann. In *IEEE Symposium on Artificial Life (IEEE ALIFE 2011)*, pages 62–69. IEEE.
- Hamann, H., Schmickl, T., Wörn, H., and Crailsheim, K. (2011b). Analysis of emergent symmetry breaking in collective decision making. *Neural Computing & Applications*. in press.
- Hereford, J. M. (2011). Analysis of BEECLUST swarm algorithm. In *Proc. of the IEEE Symposium on Swarm Intelligence (SIS 2011)*, pages 192–198. IEEE.
- Maxwell, J. C. (1871). *Theory of Heat*. Dover Publications.
- Maxwell, J. C. (1878). Tait’s ‘Thermodynamics’ (I). *Nature*, 17:257–259.
- Nicolis, G. and Prigogine, I. (1977). *Self-organization in nonequilibrium systems*. Wiley New York.
- Parunak, H. V. D. and Brueckner, S. (2001). Entropy and self-organization in multi-agent systems. In *AGENTS’01: Proceedings of the fifth international conference on Autonomous agents*, pages 124–130, New York, NY, USA. ACM Press.
- Polani, D. (2008). Foundations and formalizations of self-organization. In Prokopenko, M., editor, *Advances in Applied Self-organizing Systems*, Advanced Information and Knowledge Processing. Springer-Verlag.
- Prigogine, I. (1997). *The End of Certainty: Time, Chaos, and The New Laws of Nature*. Free Press.
- Schmickl, T. and Hamann, H. (2011). BEECLUST: A swarm algorithm derived from honeybees. In Xiao, Y. and Hu, F., editors, *Bio-inspired Computing and Communication Networks*. Routledge.
- Schmickl, T., Hamann, H., Wörn, H., and Crailsheim, K. (2009). Two different approaches to a macroscopic model of a bio-inspired robotic swarm. *Robotics and Autonomous Systems*, 57(9):913–921.
- Schmickl, T., Thenius, R., Möslinger, C., Radspieler, G., Kernbach, S., and Crailsheim, K. (2008). Get in touch: Cooperative decision making based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems*, 18(1):133–155.
- Schweitzer, F. (2003). *Brownian Agents and Active Particles. On the Emergence of Complex Behavior in the Natural and Social Sciences*. Springer-Verlag, Berlin, Germany.
- Smith, E. (2008). Thermodynamics of natural selection I: Energy flow and the limits on organization. *Journal of Theoretical Biology*, 252(2):185197.
- Szopek, M., Radspieler, G., Schmickl, T., Thenius, R., and Crailsheim, K. (2008). Recording and tracking of locomotion and clustering behavior in young honeybees (*Apis mellifera*). In Spink, A., Ballintijn, M., Bogers, N., Grieco, F., Loijens, L., Noldus, L., Smit, G., and Zimmermann, P., editors, *Proceedings of Measuring Behavior 2008, Maastricht, August 26-29*, volume 6, page 327.
- Wang, G. M., Seveck, E. M., Mittag, E., Searles, D. J., and Evans, D. J. (2002). Experimental demonstration of violations of the second law of thermodynamics for small systems and short time scales. *Phys. Rev. Lett.*, 89(5):050601.
- Weeks, J. D., Chandler, D., and Andersen, H. C. (1971). Role of repulsive forces in determining the equilibrium structure of simple liquids. *Journal of Chemical Physics*, 54(12):5237.
- Wu, M., Ahlers, G., and Cannell, D. (1995). Thermally induced fluctuations below the onset of Rayleigh-Bénard convection. *Physical review letters*, 75(9):1743–1746.

# Global Structure of Directed Networks Emerging from a Category Theoretical Formulation of the Idea “Objects as Processes, Interactions as Interfaces”

Taichi Haruna<sup>1,2</sup>

<sup>1</sup>Department of Earth & Planetary Sciences, Graduate School of Science, Kobe University,  
1-1, Rokkodaicho, Nada, Kobe, 657-8501, Japan

<sup>2</sup>PRESTO, Japan Science and Technology Agency (JST), 4-1-8 Honcho Kawaguchi, Saitama, Japan  
tharuna@penguin.kobe-u.ac.jp

## Abstract

A system of interacting elements can be represented by a directed network so that elements are nodes and interaction between two elements is an arc. Conventionally, each node is just a point, each arc represents some kind of interaction between two nodes and nothing more after the system is mapped to a directed network. However, in many real systems, each element has its own intra-node process and interaction between two elements can be seen as an interface between two intra-node processes. We can formalize this idea “objects as processes, interactions as interfaces” within the framework of category theory. We show that a new notion of connectedness called *lateral connectedness* emerges as a canonical structure obtained from the idea. Lateral connectedness is not defined on the set of nodes of a directed network, but on the set of arcs. By its definition, it may be associated with functional commonality between arcs emerging from shared input or output. As a first application, we examine significance of lateral connectedness in the neuronal network of the nematode *Caenorhabditis elegans* by comparing the partition of the set of arcs induced by the connectedness and the partitions based on neuron functions. Lateral connectedness can capture a part of functional segregation of the neuronal network above a certain interaction strength level.

## Introduction

Science of complex networks is one promising approach to understand the intrinsic organization of living systems (Alon, 2006; Junker and Schreiber, 2008; Sporns, 2011). Many characteristics such as degree distributions, average path length, clustering coefficients, centralities, assortativity coefficient, network motifs have been introduced in order to reveal functionality of biological, social, technological systems from network topology (Boccaletti et al., 2006; Newman et al., 2006; Newman, 2010). These characteristics are based on the idea which I would like to call the *real view* on networks: each node is just a point and edges or arcs between nodes indicate the existence of some kind of interaction between nodes if a system is represented as a network. However, in many real systems, it is the case that some kind of process is running within an object represented by a node. For example, in neuronal networks, nodes are neurons that have information processing ability.

In gene regulation networks, nodes are genes, but we should include proteins coded by those genes into nodes if we consider regulation relationships as arcs. Thus, we can think that complicated chemical processes to synthesize proteins occur within each node in a gene regulation network. We can interpret other biological networks including ecological networks, metabolic networks in the same way. If we consider objects as processes, then interactions between objects can be seen as interfaces between processes. I would like to call this view “objects as processes, interactions as interfaces” on networks the *dual view* in contrast to the ordinary real view mentioned above.

In this paper, we examine what is involved in having internal processes on nodes in general for complex networks. Usually, processes occurring on nodes are described as particular dynamics. Then, an appropriate statistical ensemble of dynamics is studied in order to conclude something in general (e.g. random Boolean networks by (Kauffman, 1969)). Instead of statistical generality, we here appeal to category theoretical universality to study the problem.

We note that there is an inverse dual view, namely, “processes as objects”. This idea appears in the formulation of Metabolism-Repair System by R. Rosen (Rosen, 1958). Recently, the idea was used as the line graph transformation in the community detection problem in complex networks (Ahn et al., 2010; Evans and Lambiotte, 2009). The two ideas “objects as processes” and “processes as objects” have a certain dual relationship called category theoretical adjunction (MacLane, 1998) if they are formalized within the framework of category theory (Haruna and Gunji, 2007; Pultr, 1979).

There are many ways (indeed, uncountably many ways) to consider objects as processes. However, we can show that there exists a canonical way (in a precise mathematical sense stated in Section 3) among all the ways to see objects as processes within the framework of category theory (Haruna, 2011b). The canonical way to see objects as processes gives rise to an equivalence relation on the set of arcs of each directed network. This equivalence relation can be interpreted as defining a new notion of connectedness called *lateral con-*



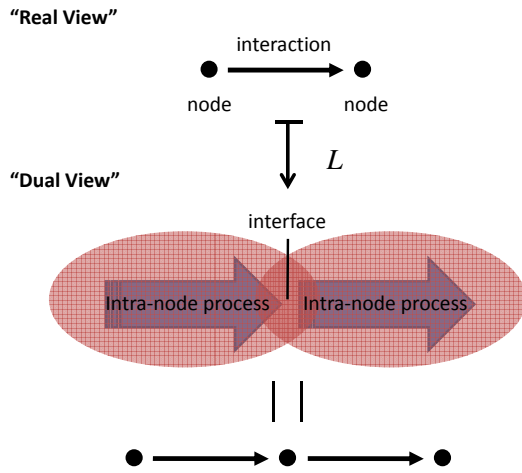


Figure 1: The idea “objects as processes, interactions as interfaces”.

*nectedness*. An intuitive explanation of derivation of lateral connectedness without category theory is the main aim of the former half of this paper. In the latter half, we analyze the neuronal network of the nematode *Caenorhabditis elegans* based on lateral connectedness as a first application to real world networks.

This paper is organized as follows. In section 2, we describe a mathematical formulation of the dual view on directed networks. In section 3, we introduce lateral connectedness for directed networks as a naturally emerging structure from the dual view. In section 4, we apply lateral connectedness to the neuronal network of *C. elegans* and discuss its functional significance. In section 5, we give conclusions and outlooks.

## Objects as Processes, Interactions as Interfaces

In this paper, we only consider directed networks. Some early attempts related to the content of this section are found in Haruna and Gunji (2007); Haruna (2008a,b, 2011a).

In the dual view introduced in Section 1, each node is interpreted as a process and each arc is seen as an interface between two processes. This idea can be formalized as network transformations as follows.

As a motivating example, let us interpret each node as an arc (together with its source and target nodes) representing a process running in the node and each arc as a node connecting two arcs representing processes running in the original two nodes (Fig. 1). Of course, each node can be replaced by a much more complicated network representing a process running within the node. The connection between the two complicated networks can also be arbitrary. We call a network (that can be arbitrary complicated) representing a process running within a node together with information

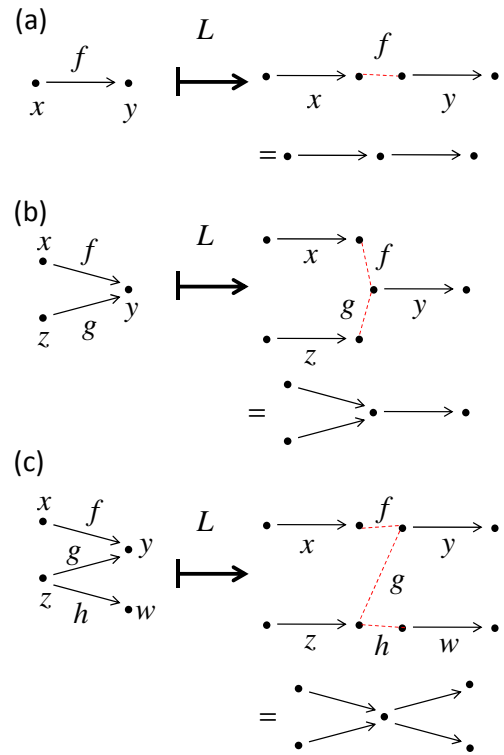


Figure 2: Three examples of the calculation of the network transformation  $L$ .

how its two copies form an interface corresponding to an arc a *model of directed network type*. In general, models of directed network type need not consist of directed networks (Haruna, 2011b), however, in the following discussion, we restrict ourselves on models consisting of directed networks for simplicity.

Fig. 2 illustrates how the above motivating model of directed network type gives rise to a network transformation  $L$ . In Fig. 2 (a), the two nodes  $x$  and  $y$  are converted to two arcs  $x$  and  $y$  by  $L$ . The target of  $x$  and the source of  $y$  are glued by the arc  $f$  in the original network. In Fig. 2 (b), there are three copies of arcs  $x$ ,  $y$  and  $z$  after the transformation  $L$  corresponding to the three nodes  $x$ ,  $y$  and  $z$  in the original network. Their sources and targets are glued according to the arcs  $f$  and  $g$  in the original network. The similar copy and glue rule works for the example in Fig. 2 (c).

Formally, the network transformation  $L$  can be defined as follows. Let  $G = (A, O, \partial_0, \partial_1)$  be a directed network, where  $A$  is a set of arcs,  $O$  is a set of nodes and  $\partial_0$  and  $\partial_1$  are maps from the set  $A$  to the set  $O$  that send each arc to its source node and target node, respectively. The directed network  $L(G)$  obtained by the application of  $L$  to  $G$  is a quartet  $L(G) = (O, O \times \{0, 1\} / \sim, \partial'_0, \partial'_1)$ , where the set of arcs of  $L(G)$  is identical to the set of nodes  $O$  of  $G$ , the set of nodes of  $L(G)$  is a quotient set  $O \times \{0, 1\} / \sim$  and



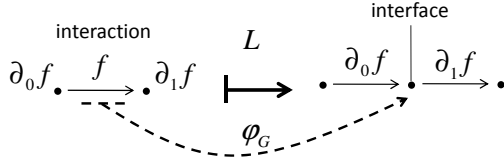


Figure 3: The map  $\varphi_G$  materialize the idea “interaction as interface”.

$\sim$  is an equivalence relation on the set  $O \times \{0, 1\}$  generated by the relation defined by  $(x, 1) \sim (y, 0)$  if and only if there exists an arc  $f$  from  $x$  to  $y$  in  $G$ . The symbol 1 indicates the “source part” of the node  $x$  and the symbol 0 indicates the “target part” of the node  $y$ . The source and target maps  $\partial'_0, \partial'_1$  are defined naturally.

In general, for any model of directed network type, the induced network transformation can be described by a similar copy and glue rule, no matter how complicated it is. For a category theoretical formulation, see (Haruna, 2011a).

### A New Notion of Connectedness

By the network transformation  $L$  introduced in Section 2, each node in a directed network  $G$  is sent to an arc in  $L(G)$ . On the other hand, we can think that each arc  $f$  in a directed network  $G = (A, O, \partial_0, \partial_1)$  is mapped to a node in  $L(G)$  between two arcs in  $L(G)$  corresponding to the source and the target nodes of  $f$ , namely,  $\partial_0 f$  and  $\partial_1 f$  (Fig. 3). We denote this map by  $\varphi_G : A \rightarrow O \times \{0, 1\} / \sim$ . For each arc  $f \in A$ ,  $\varphi_G(f)$  is the target of  $\partial_0 f$  (or the source of  $\partial_1 f$ ) in  $L(G)$ . Hence, we have  $\varphi_G(f) = [(\partial_0 f, 1)] = [(\partial_1 f, 0)]$ , where  $[(x, i)]$  is an equivalence class containing  $(x, i) \in O \times \{0, 1\}$ .

A natural question about the nature of the map  $\varphi_G$  is “When does  $\varphi_G(f) = \varphi_G(g)$  hold for arcs  $f, g \in A$ ?” The answer is straightforward and the necessary and sufficient condition for the equality  $\varphi_G(f) = \varphi_G(g)$  is that there exists a zigzag sequence of arcs between  $f$  and  $g$  as indicated in Fig. 4. We say that two arcs  $f$  and  $g$  are *laterally connected* if  $\varphi_G(f) = \varphi_G(g)$  holds.

For any model of directed network type, a similar map on the set of arcs of a given directed network can be defined. Such a map induces an equivalence relation on the set of arcs by identifying two arcs if they are sent to the same element in the codomain of the map. Let us denote the equivalence relation induced by the map  $\varphi_G$  above by  $R_{LC}$ . Then,  $R_{LC}$  is canonical in the following sense. *For any directed network  $G = (A, O, \partial_0, \partial_1)$ ,  $R_{LC}$  is the smallest equivalence relation on the set of arcs  $A$  among those induced by all models of directed network type.* In other words, the partition of the set of arcs induced by lateral connectedness is the finest one among those induced by the idea “objects as processes, interactions as interfaces”. We call each equiva-

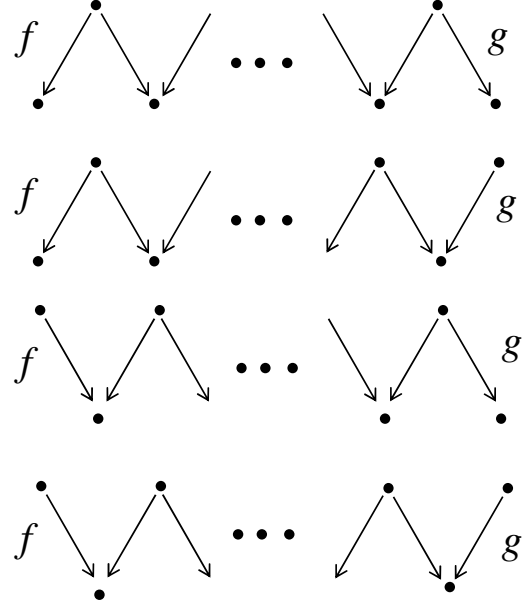


Figure 4: Two arcs  $f$  and  $g$  are laterally connected if there is a zigzag sequence of arcs between them. There are four cases depending on the situations at the both ends.

lence class *laterally connected component*. This statement can be proved within the framework of category theory in more general form (Haruna, 2011b).

In summary, we obtain the notion of lateral connectedness as a canonically emerging structure of directed networks from the idea “objects as processes, interactions as interfaces”.

By its definition, lateral connectedness may be relevant with functional commonality between arcs emerging from shared input or output. This is in contrast to the notion of strong connectedness. Here we say that two arcs are strongly connected if one arc can be reached from the other by a directed path and vice versa. Strong connectedness may be associated with functionality resulting from circulation of information or materials. Intuitively, they seem to be dual to each other. Indeed, this intuition can be enhanced by the following category theoretical point of view.

Lateral connectedness derives from the network transformation  $L$  which is based on the idea “objects as processes, interactions as interfaces”. On the other hand, strong connectedness can be obtained from the line graph transformation  $R$  which is based on the idea “processes as objects”. Given a directed network  $G = (A, O, \partial_0, \partial_1)$ , its line graph is a directed network  $R(G) = (S, A, \partial'_0, \partial'_1)$ , where  $S = \{(f, g) \in A \times A \mid \partial_1 f = \partial_0 g\}$ ,  $\partial'_0(f, g) = f$  and  $\partial'_1(f, g) = g$  for any  $(f, g) \in S$ . As noted in Section 1, the two transformations  $L$  and  $R$  satisfy a certain category theoretical duality called adjunction (Haruna and Gunji, 2007;

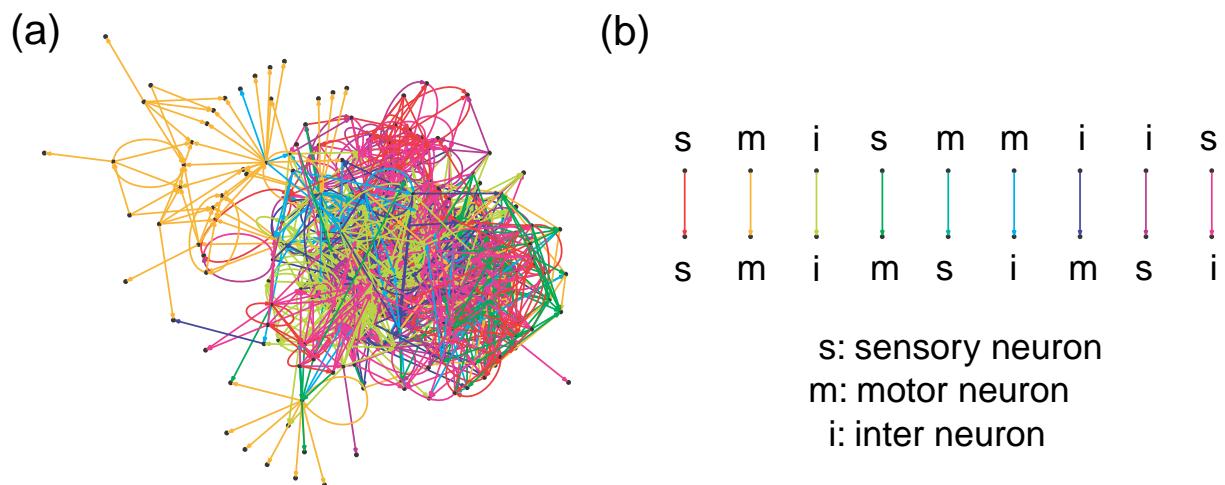


Figure 5: The wiring diagram of the neuronal network of *C. elegans* based on the database (Oshio et al., 2003) (<http://ims.dse.ibaraki.ac.jp/ccep/>) depicted by Graphviz (<http://www.graphviz.org/>). (a) Arcs are colored based on pairs of functions of their source and target neurons. (b) Correspondence between colors and pairs of functions of neurons.

Pultr, 1979). By definition,  $S$  is the set of arcs of the directed network  $R(G)$ , but it can be seen as a binary relation on the set  $A$ . Mathematically, the notion of strong connectedness defined above is an equivalence relation  $SC$  on the set  $A$  of arcs of  $G$ . On the other hand, we have an equality

$$SC = \overline{S} \cap \overline{S^{-1}}, \quad (1)$$

where  $S^{-1}$  is the inverse of the binary relation  $S$  and  $\overline{T}$  for a binary relation  $T$  on  $A$  is its reflexive and transitive closure. In this sense, strong connectedness is generated by the line graph transformation  $R$  which is category theoretical dual to  $L$ .

One might think that the duality between lateral connectedness and strong connectedness in the above sense is a mathematical expression for Lorente de Nó's two principles of plurality and reciprocity (Lorente de Nó, 1938).

## Analysis of a Neuronal Network

In this section, we discuss significance of lateral connectedness in the neuronal network of *C. elegans* as a first application of it. We compare the partitions of the set of arcs based on functions of neurons with the partition induced by lateral connectedness to examine functional significance of lateral connectedness. We make use of two similarity measures described in the next subsection for the comparison.

### Network Data

We make use of the database constructed by Oshio et al. (2003) (<http://ims.dse.ibaraki.ac.jp/ccep/>) whose original reference is White et al. (1986). We remove nodes and connections other than neurons and chemical synapses. The resulting data set contains 233 neurons among 282 neurons

in the somatic nervous system and 4170 chemical synapses. We construct a family of directed networks whose nodes are 233 neurons in the following way: First, we put an arc from one node to another node if there exists a chemical synapse from the former to the latter. Second, since there is multiple chemical synapses from one neuron to another neuron in general, we specify a weight for each arc by the number of chemical synapses from the source to the target of the arc. Finally, we introduce thresholds for the weight values and consider the network topology consisting of arcs whose weights are greater or equal to a given threshold.

Each neuron has one of three functional types: sensory, inter and motor. We consider three partitions of the set of arcs based on the functions of neurons. The first one is called *ST-partition* which considers types of the two neurons at both ends of each arc. Thus, there are nine clusters for the ST-partition. In the wiring diagram shown in Fig. 5 (a) where threshold is 1, each arc is colored based on the ST-partition. The correspondence between colors and the ST-partition clusters is indicated in Fig. 5 (b). The second one is called the *S-partition* which considers type of the source neuron of each arc. The third one is called the *T-partition* which considers type of the target neuron of each arc. The number of clusters in the S-partition or T-partition is three.

The equivalence relation  $R_{LC}$  induced by lateral connectedness also gives rise to a partition of the set of arcs. We call this partition the *LC-partition*. In the following discussion, we measure similarity between the LC-partition and the above three functional partitions.

### Similarity Measures

We make use of two similarity measures to quantify similarity between two partitions on a set. The first one is called the

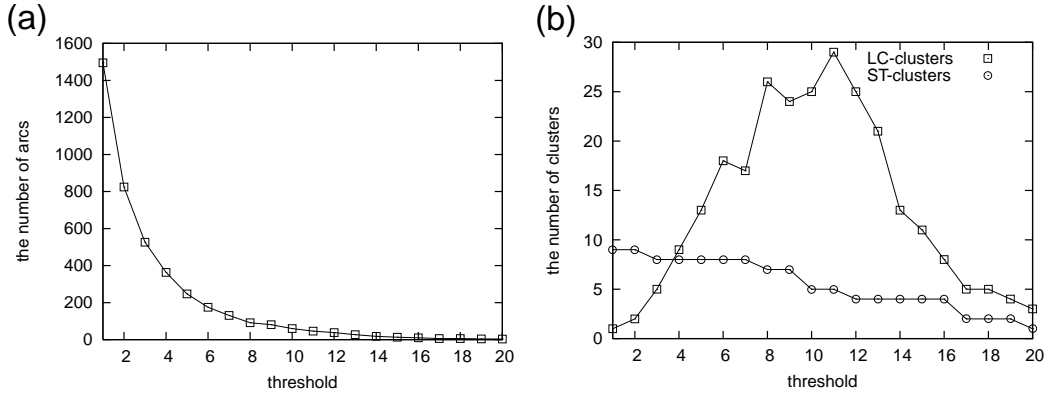


Figure 6: (a) The number of arcs as a function of threshold. (b) The number of clusters in the LC-partition and in the ST-partition as a function of threshold.

*Adjusted Rand Index (ARI)* (Hubert and Arabie, 1985). The second one is called the *Adjusted normalized Mutual Information (AMI)* (Vinh et al., 2009). To explain the idea of the ARI, we first review the definition of the *Rand Index (RI)* (Rand, 1971).

Let  $X$  be a set consisting of  $N$  points. Let  $\mathbf{U} = \{U_1, U_2, \dots, U_l\}$  and  $\mathbf{V} = \{V_1, V_2, \dots, V_m\}$  be two partitions of  $X$ , namely, they are families of subsets of  $X$  satisfying  $U_i \cap U_{i'} = V_j \cap V_{j'} = \emptyset$  for  $i \neq i', j \neq j'$  and  $\cup_{i=1}^l U_i = \cup_{j=1}^m V_j = X$ . Let us put  $n_{ij} := |U_i \cap V_j|$ ,  $a_i := |U_i|$  and  $b_j := |V_j|$  for  $i = 1, 2, \dots, l$  and  $j = 1, 2, \dots, m$ , where  $|Y|$  for a set  $Y$  denotes its cardinality. Then, we have  $a_i = \sum_{j=1}^m n_{ij}$  and  $b_j = \sum_{i=1}^l n_{ij}$  for  $i = 1, 2, \dots, l$  and  $j = 1, 2, \dots, m$ . A  $l \times m$  matrix  $C := (n_{ij})$  is called the *contingency matrix*, which encodes information how two partitions  $\mathbf{U}$  and  $\mathbf{V}$  overlap. We can calculate both the ARI and the AMI by using elements of the contingency matrix  $C$ .

The Rand Index (RI) between partitions  $\mathbf{U}$  and  $\mathbf{V}$  is defined by counting the number of pairs of elements of  $X$  on which two partitions agree or disagree:

$$RI(\mathbf{U}, \mathbf{V}) = \frac{N_{00} + N_{11}}{N_{00} + N_{01} + N_{10} + N_{11}}, \quad (2)$$

where  $N_{00}$  is the number of pairs that are in the same cluster in both  $\mathbf{U}$  and  $\mathbf{V}$ ,  $N_{01}$  is the number of pairs that are in the same cluster in  $\mathbf{U}$  but in different clusters in  $\mathbf{V}$ ,  $N_{10}$  is the number of pairs that are in different clusters in  $\mathbf{U}$  but in the same cluster in  $\mathbf{V}$  and  $N_{11}$  is the number of pairs that are in different clusters in both  $\mathbf{U}$  and  $\mathbf{V}$ . After a few algebras, one can see that  $N_{01}$  and  $N_{10}$  are given by

$$N_{01} = \sum_{j=1}^m \binom{b_j}{2} - \sum_{i=1}^l \sum_{j=1}^m \binom{n_{ij}}{2}, \quad (3)$$

$$N_{10} = \sum_{i=1}^l \binom{a_i}{2} - \sum_{i=1}^l \sum_{j=1}^m \binom{n_{ij}}{2}. \quad (4)$$

Since we have  $N_{00} + N_{01} + N_{10} + N_{11} = \binom{N}{2}$ , we obtain the following explicit formula for the RI:

$$RI(\mathbf{U}, \mathbf{V}) = \frac{\binom{N}{2} - \left\{ \sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2} \right\} + 2 \sum_{i,j} \binom{n_{ij}}{2}}{\binom{N}{2}}. \quad (5)$$

The RI takes its maximum value 1 when two partitions are identical. The minimum value 0 is taken if and only if one partition consists of a single cluster and the other consists of only clusters with a single point, which is hard to satisfy by random partitions. Indeed, the RI takes relatively high values for two random partitions. However, it is plausible for a similarity measure to take values close to zero for random partitions. To improve this disadvantage of the RI, Hubert and Arabie (1985) introduced the Adjusted Rand Index (ARI) which takes over a correction for chance:

$$ARI(\mathbf{U}, \mathbf{V}) = \frac{RI(\mathbf{U}, \mathbf{V}) - E(RI|\mathbf{a}, \mathbf{b})}{1 - E(RI|\mathbf{a}, \mathbf{b})}, \quad (6)$$

where 1 in the denominator is the maximum value of the RI and  $E(RI|\mathbf{a}, \mathbf{b})$  is the expected value of the RI between two randomly chosen partitions of the set  $X$  subject to the condition that two vectors  $\mathbf{a} = (a_1, a_2, \dots, a_l)$  and  $\mathbf{b} = (b_1, b_2, \dots, b_m)$  are fixed. Since we have  $E\left(\binom{n_{ij}}{2} | \mathbf{a}, \mathbf{b}\right) = \binom{a_i}{2} \binom{b_j}{2} / \binom{N}{2}$  (Hubert and Arabie, 1985), an explicit formula for the ARI is given by

$$ARI(\mathbf{U}, \mathbf{V}) = \frac{\sum_{i,j} \binom{n_{ij}}{2} - \left\{ \sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right\} / \binom{N}{2}}{\frac{1}{2} \left\{ \sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2} \right\} - \left\{ \sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right\} / \binom{N}{2}}. \quad (7)$$

Our second measure of similarity, the AMI is defined based on the mutual information between two partitions (Vinh et al., 2009). Let us introduce the probability that an element of  $X$  is contained in a cluster  $U_i$  by  $P(i) = a_i/N$ .

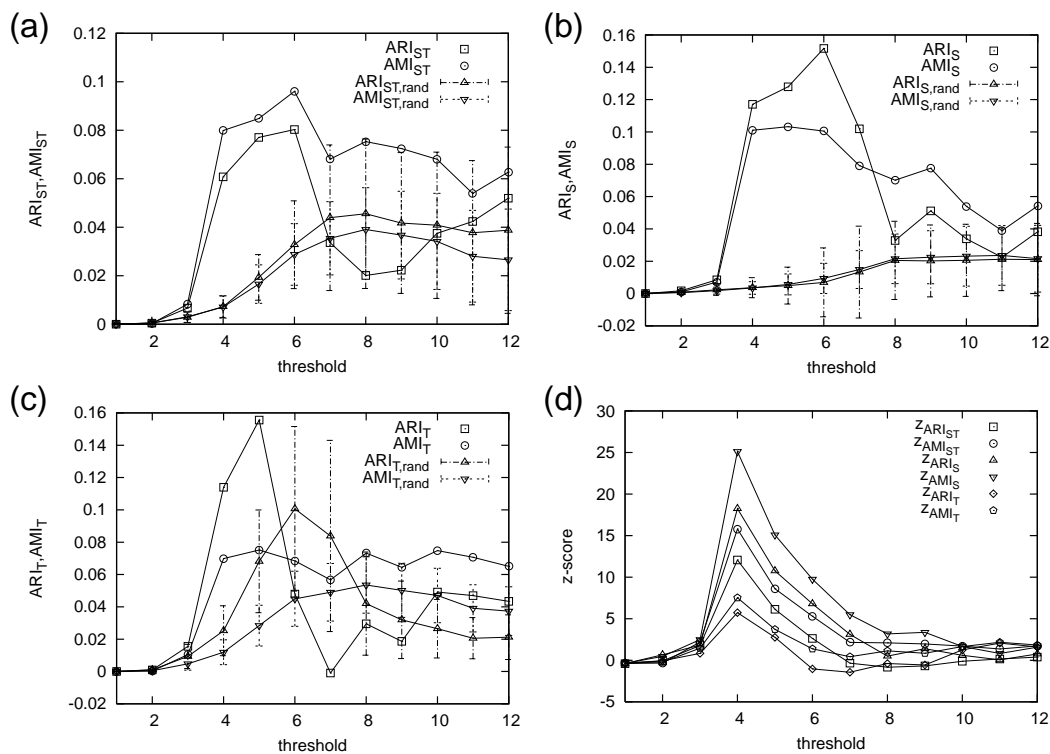


Figure 7: Comparisons between the LC-partition and partitions based on neuron functions. Two similarity measures, one is pair-counting based (the ARI) and the other is information-theoretic (the AMI), are used. (a) The LC-partition vs the ST-partition. (b) The LC-partition vs the S-partition. (c) The LC-partition vs the T-partition. (d) Z-scores for the values of two similarity measures as functions of threshold.

The Shannon entropy with respect to the partition  $\mathbf{U}$  is defined by  $H(\mathbf{U}) = -\sum_{i=1}^l P(i) \log_2 P(i)$ . Similarly, the Shannon entropy with respect to the partition  $\mathbf{V}$  is given by  $H(\mathbf{V}) = -\sum_{j=1}^m P'(j) \log_2 P'(j)$ , where  $P'(j) = b_j/N$ . Then, the mutual information between two partitions  $\mathbf{U}$  and  $\mathbf{V}$  is defined by  $I(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^l \sum_{j=1}^m P(i, j) \log_2 \frac{P(i, j)}{P(i)P'(j)}$ , where  $P(i, j) = n_{ij}/N$  which is the joint probability that an element of  $X$  falls into both  $U_i$  and  $V_j$ .

Strehl and Ghosh (2002) proposed the normalized mutual information (NMI) as follows:

$$NMI(\mathbf{U}, \mathbf{V}) = \frac{I(\mathbf{U}, \mathbf{V})}{\sqrt{H(\mathbf{U})H(\mathbf{V})}}, \quad (8)$$

which takes its values in the unit interval  $[0, 1]$ . The NMI takes its maximum value 1 when two partitions are identical. The minimum value 0 is realized when two partitions are independent, namely,  $n_{ij} = a_i b_j$  holds for all  $1 \leq i \leq l$  and  $1 \leq j \leq m$ . Hence, the NMI for random partitions takes its values close to 0. However, its adjusted version is more preferable. The adjusted normalized mutual information (AMI) is defined in the similar spirit as in the ARI (Vinh

et al., 2009) :

$$AMI(\mathbf{U}, \mathbf{V}) = \frac{I(\mathbf{U}, \mathbf{V}) - E(I|\mathbf{a}, \mathbf{b})}{\sqrt{H(\mathbf{U})H(\mathbf{V}) - E(I|\mathbf{a}, \mathbf{b})}}, \quad (9)$$

where  $E(I|\mathbf{a}, \mathbf{b})$  is the expected value of the mutual information  $I$  between two randomly chosen partitions of the set  $X$  subject to the condition that two vectors  $\mathbf{a} = (a_1, a_2, \dots, a_l)$  and  $\mathbf{b} = (b_1, b_2, \dots, b_m)$  are fixed.

In the next subsection, we apply these two adjusted similarity measures, the ARI and the AMI, to the partitions of the set of arcs in the neuronal network of *C. elegans* by neuron functions and the partition based on lateral connectedness for each threshold.

## Results

Fig. 6 (a) shows the number of arcs as a function of threshold. Fig. 6 (b) indicates the number of clusters in the LC-partition and in the ST-partition. The former tends to increase for thresholds within the range from 1 to 10 because decrease in the number of arcs can lead to division of one cluster into two or more clusters. It decreases for thresholds larger than 12 simply because the number of arcs is too small

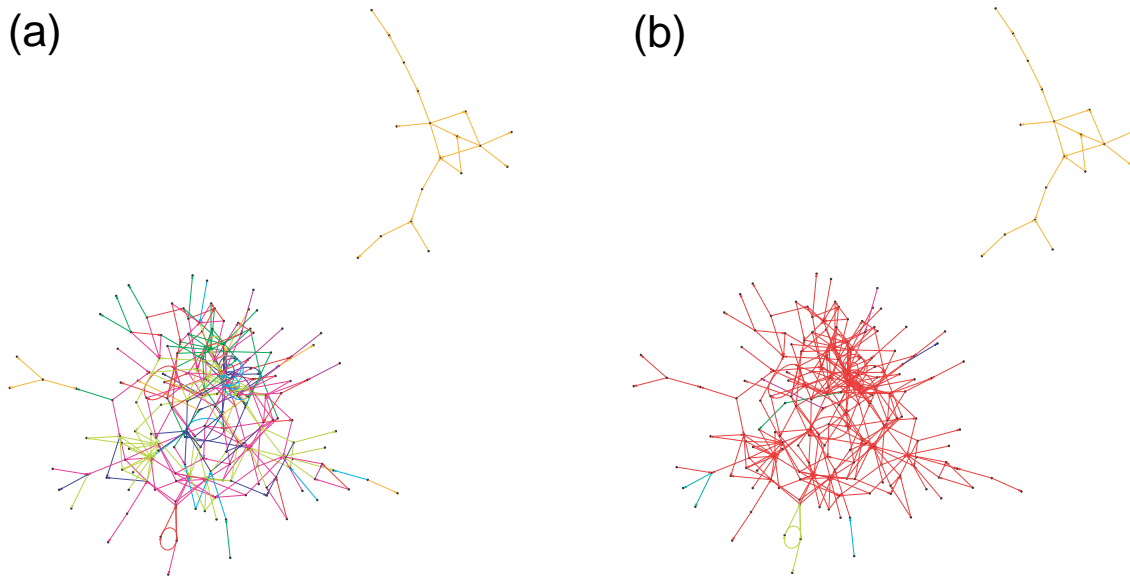


Figure 8: The wiring diagram of the neuronal network of *C. elegans* at threshold 4 depicted by Graphviz (<http://www.graphviz.org/>). (a) Arcs are colored by the ST-partition. Color assignment is the same for Fig. 5. (b) Arcs are colored by the LC-partition. Colors of different clusters are specified arbitrarily.

for the number of the LC-clusters to grow by divisions for those relatively large thresholds.

In Fig. 7 (a), we plot the ARI and the AMI between the LC-partition and the ST-partition as a function of threshold. It takes its maximum value when threshold is equal to 6. As a control experiment, we calculate averages and standard deviations of the ARI and the AMI between the LC-partition and the ST-partition on 1000 randomized networks by re-wiring arcs randomly, which are also shown in Fig. 7 (a). Note that degree distributions are invariant under the re-wiring process. We can see a large deviation from the control around the maximum point. We have similar results for the S-partition and the T-partition (Fig. 7 (b),(c)).

To quantify deviation from the control experiment, we calculate the Z-score for each comparison. The Z-score of a quantity  $Q$  is defined by

$$z_Q = \frac{Q_{orig} - \langle Q_{rand} \rangle}{\sigma}, \quad (10)$$

where  $Q_{orig}$  is the value of  $Q$  in the original network,  $\langle Q_{rand} \rangle$  is the average of  $Q$  calculated from an ensemble of randomized networks and  $\sigma$  is its standard deviation. The Z-scores of both the ARI and the AMI take their maximum value when threshold is equal to 4 for all comparisons (Fig. 7 (d)). All of the maximum values of the Z-scores are more than 5, which indicates significant deviation from the control in all comparisons. However, we should note that the absolute values of the two similarity measures are not so high, at most 0.152.

In Fig. 8, we plot the wiring diagram where we set thresh-

old 4. Arcs are colored based on the ST-partition (Fig. 8 (a)) and the LC-partition (Fig. 8 (b)). There are two weakly connected components, one is large and the other is small. Here, we define a weakly connected component of a directed network as a maximal set of arcs in which every pair of arcs are connected by a sequence of arcs ignoring the direction. Almost all the motor-motor connections are included in the smaller weakly connected component on one hand, they also form a single laterally connected component in the LC-partition. However, the LC-partition fails to capture more detailed functional partition within the larger weakly connected component possibly due to many recurrent connections between the sensor, inter and motor region of the neuronal network of *C. elegans* (Varshney et al., 2011). This is one reason that we have relatively low absolute values for the two similarity measures.

## Conclusions and Outlooks

In this paper, we intuitively explained that how the idea “objects as processes, interactions as interfaces” can be formalized within the framework of category theory. We derived the notion of lateral connectedness as a canonical structure obtained from the idea. By its definition, lateral connectedness has possibility to be associated with functional commonality between arcs arising from shared input or output. As a first application of lateral connectedness, we examined functional significance of lateral connectedness in the neuronal network of *C. elegans* by the method of clustering comparison. For the analysis, we made use of two similarity measures to quantify similarity between two partitions

on the same set, one is pair-counting based measure and the other is information-theoretic measure.

We showed that the partition of the set of arcs based on lateral connectedness is not inconsistent with the functional partition of the set of arcs. However, even if we set threshold at the point where the largest deviation from an ensemble of randomized networks is observed, it can only capture a part of the partitions based on neuron functions. One problem of the analysis performed in this paper may be that the direct comparison to functional partitions is too strict to recognize significance of lateral connectedness. Another problem is that the data used is incomplete. Analysis with more complete data (Varshney et al., 2011) will be necessary.

Introduction of lateral connectedness has several implications. First, we can analytically solve percolation problems with respect to lateral connectedness on configuration model (networks chosen uniformly at random from the set of all possible networks with a specified degree distribution) of directed networks (Haruna, 2011b). Applications of the analytical result on configuration model to biological networks are now ongoing. Second, we can define alternatives for some notions used in conventional complex network studies. For example, the notion of path length can be defined based on lateral connectedness. Since metrics such as closeness and betweenness centralities are functions of path lengths, they are also the targets of alternative definitions. Finally, theoretical development and empirical applications of the duality between lateral connectedness and strong connectedness are also intriguing issues.

## Acknowledgements

This work was supported by JST PRESTO program.

## References

- Ahn, Y.-Y., Bagrow, J. P., and Lehmann, S. (2010). Link communities reveal multiscale complexity in networks. *Nature*, 466:761–765.
- Alon, U. (2006). *Introduction to Systems Biology: Design Principles of Biological Circuits*. CRC Press, Boca Raton.
- Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D.-U. (2006). Complex networks: Structure and dynamics. *Phys. Rep.*, 424:175–308.
- Evans, T. S. and Lambiotte, R. (2009). Line graphs, link partitions, and overlapping communities. *Phys. Rev. E*, 80:016105.
- Haruna, T. (2008a). Algebraic theory of biological organization. Doctoral Dissertation, Kobe University.
- Haruna, T. (2008b). Being arranged in advance: Quantum entanglement and biological feedback. In Bullock, S., Noble, J., Watson, R., and Bedau, M. A., editors, *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 220–226. MIT Press, Cambridge, MA.
- Haruna, T. (2011a). An application of category theory to the study of complex networks. *Int. J. Comp. Anti. Sys.* (in press).
- Haruna, T. (2011b). In preparation.
- Haruna, T. and Gunji, Y.-P. (2007). Duality between decomposition and gluing: A theoretical biology via adjoint functors. *BioSystems*, 90:716–727.
- Hubert, L. and Arabie, P. (1985). Comparing partitions. *J. Classification*, 2:193–218.
- Junker, B. H. and Schreiber, F. (2008). *Analysis of Biological Networks*. John Wiley & Sons, Inc.
- Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, 22:437–467.
- Lorente de N6, R. (1938). Analysis of the activity of the chains of internuncial neurons. *J. Neurophysiol.*, 1:207–244.
- MacLane, S. (1998). *Categories for the Working Mathematician, 2nd edition*. Springer-Verlag, New York.
- Newman, M. E. J. (2010). *Networks: An Introduction*. Oxford Univ. Press Inc., New York.
- Newman, M. E. J., Barabási, A.-L., and Watts, D. J. (2006). *The Structure and Dynamics of Networks*. Princeton Univ. Press, Princeton.
- Oshio, K., Iwasaki, Y., Morita, S., Osana, Y., Gomi, S., Akiyama, E., Omata, K., Oka, K., and Kawamura, K. (2003). Database of synaptic connectivity of *C. elegans* for computation. Technical Report of CCEP, Keio Future, No. 3, Keio University.
- Pultr, A. (1979). On linear representations of graphs. In *Fundamentals of computation theory (Proc. Conf. Algebraic, Arith. and Categorical Methods in Comput. Theory, Berlin/Wendisch-Rietz, 1979)*, *Math. Res.* 2, pages 362–369. Akademie-Verlag, Berlin.
- Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.*, 66:846–850.
- Rosen, R. (1958). The representation of biological systems from the standpoint of the theory of categories. *Bull. Math. Biophys.*, 20:317–341.
- Sporns, O. (2011). *Networks of the Brain*. MIT Press, Cambridge, MA.
- Strehl, A. and Ghosh, J. (2002). Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617.
- Varshney, L. R., Chen, B. L., Paniagua, E., Hall, D. H., and Chklovskii, D. B. (2011). Structural properties of the *Caenorhabditis elegans* neuronal network. *PLoS computational biology*, 7:e1001066.
- Vinh, N. X., Epps, J., and Bailey, J. (2009). Information theoretic measures for clusterings comparison: is a correction for chance necessary? In Danyluk, A., Bottou, L., and Littman, M., editors, *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1073–1080. ACM New York, NY, USA.
- White, J. G., Southgate, E., Thomson, J. N., and Brenner, S. (1986). The structure of the nervous system of the nematode *Caenorhabditis elegans*. *Phil. Trans. R. Soc. London B*, 314:1–340.

# Opening Stable Doors: Complexity and Stability in Nonlinear Systems

Inman Harvey<sup>1</sup>

<sup>1</sup>Evolutionary and Adaptive Systems Group, University of Sussex  
inmanh@gmail.com

## Abstract

Generic complex systems of many interacting parts can model both natural and artificial systems, and the conditions for their stability are of interest. Two influential papers (Gardner and Ashby, 1970; May, 1972) laid down a mathematical framework suggesting that, without some specific constraints on the interactions, such systems are very likely to be unstable as they increase in size and connectance. We draw attention to a programming error in the first paper and to flaws and omissions in reasoning in the second that discredit such conclusions when applied to nonlinear systems. With nonlinearity the connectance strength of an influence of any one variable upon any other will vary according to context, which May's analysis does not address. Further, in nonlinear systems there can be many equilibria, and global instability requires every relevant local equilibrium to be unstable; neglecting this invalidates the conclusions. We discuss the relevance of ambiguous circuits (Thomas and D'Ari, 1990) and consider simple classes of nonlinear functions that generate these, including the hat shaped viability functions that generate homeostasis in Daisyworld models. We demonstrate that the May results are unreliable even for the simplest families of nonlinear systems that model common biological, physical or artificial systems.

## Introduction

An influential early paper (Gardner and Ashby, 1970) used computer simulations to assess the probability that a large system of interacting component parts that has been assembled at random, or has grown haphazardly, will be stable or unstable. They considered systems where the interactions between parts were linear, and looked at how the expectation of stability changed as the number of variables increased. This was a theoretical study, to be motivated by its possible application to both biological and man-made systems: brains (real or artificial), planetary climate systems, social or financial systems, ecosystems. The conclusion was the suggestion that all such large (random or haphazard) complex linear dynamic systems may be expected to show the property of being stable up to some critical, fairly small, level of connections; but above that phase transition value they are overwhelmingly likely to be unstable. From this it could be deduced that if one observed large complex linear systems that were indeed stable, there must be something exceptional and non-random about the way that the parts were connected.

The influence of this work stems primarily from its extension and development by Robert May, and the subsequent proliferation of a wide body of research in this

area. He replicated a version of the results analytically rather than computationally (May, 1972), and claimed that their validity extended beyond the linear systems of Gardner and Ashby (hereafter: G&A) to systems "which in general may obey some quite nonlinear set of first-order differential equations". May's interest mainly focused on ecological systems, and a subsequent book (May, 1973) largely set the agenda for discussion of the relationship between complexity and stability in ecosystems ever since.

Before this work there was a common perception that the more diverse was the range of species in an ecosystem, the more robust and resilient to perturbations that system would be; and further, it was often assumed that this may well be due to some underlying law of large numbers that could apply very generally across all sorts of systems with many interacting components. But the work of G&A and May, apparently using very minimal mathematical assumptions, appeared to suggest that the opposite was true – at least, in the absence of further specific constraints. So subsequent argument and analysis have tended to focus on what further constraints, what limitations on the number, sign and size of interspecies interactions, might be necessary in order to make it likely that a complex ecosystem was stable. The mathematics, it has largely been assumed, is relatively simple and correct. Hence if we want to explain the existence of complex stable systems, it looks like we need to add further assumptions.

In this paper, we shall demonstrate that the reasoning within these two primary sources (Gardner and Ashby, 1970, May, 1972) is partially invalidated through omission and errors, and in particular should not be generalised in this way to nonlinear systems. Firstly, we draw attention to a programming error in the G&A paper, which has been noted previously (Solow et al., 1999). Secondly, we point out that May's attempted extension to nonlinear systems fails to specify the distribution from which the relevant connection strengths are drawn.

Thirdly, and fatally to May's reasoning, we point out a flaw where he claims to go beyond the purely linear systems of G&A towards a more general set of nonlinear systems. May considers local stability at just a single fixed point in the space of possible values for the system, a point that makes sense when considering linear systems with negative self-interactions. Unfortunately, when we move on to nonlinear systems there can be a large (and in some circumstances unlimited) number of points of potential stability to consider. Global instability would require local instability at every one of those points. Hence the probability of global stability will



be underestimated if one just considers local stability at a single fixed point, as May does.

These various criticisms are, as far as we are aware, all drawn together here for the first time. We present examples demonstrating that it is not merely exotic nonlinear functions that raise these issues. Even simple monotonic nonlinear functions such as sigmoids, or the simplest piecewise linear functions with a single change of slope, are sufficient to invalidate the reasoning. Hat shaped viability functions, as used in Daisyworld models, are discussed and it is shown how stability arises independently of the sign of the opposing effect. The ‘ambiguous circuits’ so produced are related to the multistationarity analysis of Thomas (Thomas and Kaufman, 2001a, 2001b).

These flaws in the two foundational papers by Gardner and Ashby (1970) and May (1972) suggest that a radical reappraisal is needed in the mathematical foundations of a substantial body of work that has built up over some 40 years. Rather than seeking a route to stability by adding further constraints to these abstract models, we need to open the doors to those possible locations of stability that have until now, through error or omission, been excluded. The significance goes beyond ecosystem theory to the study of all kinds of natural and artificial systems with complex nonlinear interactions, including financial systems (Haldane and May, 2011).

## Gardner and Ashby on Linear Systems

Their short paper, a Letter of less than one page in *Nature* (Gardner and Ashby, 1970), was an early example of a computer simulation, using a Monte Carlo approach. They considered a very simplified formal model of any large system of many interacting parts. This could be traffic at an airport, or the neurons in a human brain. They asked the question: supposing one did not know all the details of the interactions between component parts, but modelled these as coming from some random distribution that gave the signs and sizes of these interactions, then what was the chance that such a large system will be stable? Although in the real world most of these large systems, perhaps biological or social, will be grossly nonlinear, they explicitly restricted themselves to considering only systems with linear interactions, as a first step towards a more general treatment. They were interested only in fixed point equilibria.

The model had  $n$  component parts. The intention was to investigate how the generic properties of such systems varied as  $n$  increases. The instantaneous state of the system can be expressed by a vector  $\mathbf{x}$ , where  $x_i$  represents the current value of the  $i$ th variable. In the very general case of nonlinear systems we would have, with different nonlinear functions for each  $i$ :

$$\frac{dx_i}{dt} = \text{NonLinFn}_i(x_1, x_2, \dots, x_n)$$

However in this restricted linear case this simplifies to a weighted sum of the current values of all the variables:

$$\frac{dx_i}{dt} = \sum_j a_{ji} x_j \quad \text{Eqns 1}$$

Because this is a linear system, there is a unique equilibrium point where for all  $i$   $dx_i/dt=0$ . The issue will be: what is the

probability that this unique equilibrium is stable, given the distribution from which the weights  $a_{ji}$  in the connection matrix  $\mathbf{A}$  are drawn.  $\mathbf{A}$  is the Jacobian matrix of the first-order partial derivatives, and in this case of a linear system these terms are all scalars, of fixed size and sign; when later we move on to nonlinear systems, these terms will be variable in both size and sign.

G&A chose to make this a partially connected system, with a proportion  $C$  of the off-diagonal weights being nonzero. These nonzero weights were distributed evenly between -1.0 and +1.0. Further, they ensured that all the weights  $a_{ii}$  in the main diagonal of the connection matrix (self-connections) were negative. They distributed these evenly between -1.0 and -0.1; in May’s version that followed, May set all these to -1.0.

G&A are thus discussing a family of linear feedback systems, parameterised by these two values:  $n$ , the number of component parts, and  $C$ , the connectance or the proportion of possible interactions between parts that are non-zero. For any given values of  $n$  and  $C$ , their Monte Carlo approach involved testing many cases of such systems, with the connection weights drawn from the appropriate distributions, and finding out through computation what proportion of the systems were stable at their unique equilibrium point. For low values of the connectance, where the interactions are dominated by the stipulated negative values of self-connections, the probability of stability was close to 100% for all values of  $n$  tested. But as the connectance  $C$  increased, the probability of stability fell away. Using the limited computational facilities of their day (Gardner and Ashby, 1970), they tested examples where  $n$  equals 4, 7 or 10. Their conclusion, illustrated by a figure, was that as  $n$  increases the relationship between connectance and stability changes from (for  $n=4$ ) a smooth falling away of probability of stability as connectance increases towards a step function for values of  $n$  of 10 or more. Their figure (partly replicated by the thin lines in Figure 1 here) suggests that for  $n=10$  this phase transition from “almost certainly stable” to “almost certainly unstable” occurs at or around a connected value of 13%,  $C=0.13$ .

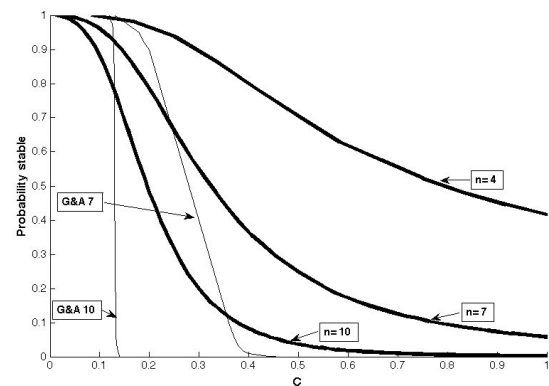


Figure 1: Thick lines give the correct results for G&A’s examples, for  $n = 10, 7, 4$  from left to right. Diagonal terms  $a_{ii}$  drawn from  $[-0.1, -1.0]$ ; a proportion  $C$  of off-diagonal terms  $a_{ji}$  ( $i \neq j$ ) drawn from  $[-1.0, 1.0]$ , with the remainder zero. Thin lines copy the incorrect results that G&A showed for  $n = 10, 7$  (Gardner and Ashby, 1970).

## The Programming Error

When we replicated their method<sup>1</sup> our results were similar for  $n=4$ , but noticeably different for  $n=7$  or 10. The difference, shown in Figure 1, is striking, and in particular eliminates the sharp nature of the phase transition claimed for  $n=10$ . This was their main result, and May claimed on the basis of his analytical treatment (May, 1972) to have corroborated this: “The sharp transition from stability to instability, which was the essential feature of their [G&A] paper is confirmed”. Having failed to replicate this sharp transition, the first step was to check whether we had misinterpreted their methods. But eventually a colleague discovered a rarely cited 1999 reference (Solow et al., 1999) pointing out the same problem, with results agreeing with our own presented here. They attributed the problem to some unknown programming error in G&A’s code. Further, they comment that this nullifies one of May’s conclusions where he had assumed that the G&A phase transition was a real phenomenon. Correction of this programming error does not alter the conclusion that as  $n$  increases and  $C$  increases the probability of stability goes down; it does alter the conclusion that for values of  $n$  above some fairly small value the relationship between stability and connectance turns into something close to a step function.

For the purposes of this paper, this programming error is the least important of the errors and omissions to be discussed. Nevertheless, it is of note that it took nearly 30 years until this error was pointed out in print.

## May’s analysis: linear systems

Whereas G&A explicitly limited themselves to the consideration of linear systems “merely as a first step towards a more general treatment” (Gardner and Ashby, 1970), May claims to be considering systems “which in general may obey some quite nonlinear set of first-order differential equations.” (May, 1972). His method is to focus on the behaviour of such nonlinear equations around “the equilibrium point”. Through making a Taylor expansion and ignoring the higher-order terms one can consider this locally as a linear system. Thereafter, May goes on to analyse the same kind of linear system as G&A, while still claiming that it generalises to nonlinear systems.

Insofar as May’s analysis is restricted to the linear version, he tackled analytically much the same class of systems that G&A had tackled computationally. To be precise, this was a slight variant with qualitatively the same behaviour; in place of just  $C$  or connectance he considers a term  $\alpha$  that is the mean square value of the distribution of all off-diagonal elements, described as expressing the average interaction “strength” (measured on a scale that rates the negative self-feedbacks on the diagonal of the matrix at -1). May’s results were broadly similar, claiming that the central feature of the results for large systems is “the very sharp transition” from stable to unstable behaviour above a critical value that “accords with Gardner and Ashby’s conjecture”. As we have pointed out above, in fact the transition is not as sharp as

G&A indicated; however the analytical results do agree with a correctly coded computational Monte Carlo approach. The influential take-home message from both the computational and analytical results has been: in any such system of many interacting parts, as soon as the average interaction strength (interactions between different component parts) rises above some small value, the probability that such a system will be stable drops to near zero. This limitation on stability becomes worse as  $n$ , the number of parts, increases. In the context of ecosystems, such a result challenges the commonly held assumption that the more diverse an ecosystem is, the better it is able to remain stable in the face of perturbations.

## Picturing Stability

In preparation for understanding nonlinear systems, we first present in some detail a sketch of how to analyse and visualise stability in linear systems. This is basic textbook material, but that is the level of the flaws that we are going to exhibit when we move on later to nonlinear systems. For a simple system of two variables, we can graphically sketch the nullclines (where  $dx/dt=0$  and  $dy/dt=0$ ) and, by plotting the consequences of perturbations, analyse for stability. We start with two linear examples, Equations 2 and 3, sketched and analysed in Figures 2 and 3:

$$\frac{dx}{dt} = -x + \frac{y}{2} + 1 \quad \frac{dy}{dt} = \frac{x}{2} - y + 1 \quad \text{Eqns 2}$$

$$\begin{pmatrix} -1 & 0.5 \\ 0.5 & -1 \end{pmatrix}$$

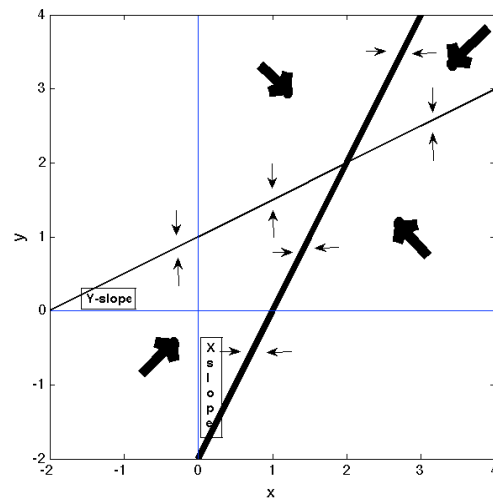


Figure 2: The nullclines for Eqns 2. Thick line for  $dx/dt=0$ , with horizontal small arrows indicating responses to  $x$ -perturbations. Thin line for  $dy/dt=0$ , with vertical small arrows for response to  $y$ -perturbations. The heavy arrows sum these responses, giving a stable equilibrium at the intersection (2,2).

<sup>1</sup> Matlab code at [www.informatics.sussex.ac.uk/users/inmanh/stable](http://www.informatics.sussex.ac.uk/users/inmanh/stable)

The Jacobian matrix restates the fact that the self-connections are -1, and the cross-interactions are 0.5. These latter correspond to  $\tan(X\text{-slope})$  and  $\tan(Y\text{-slope})$  as those angles are indicated in the figure. In contrast, consider this example with the same nullclines, though swapped around:

$$\begin{aligned} \frac{dx}{dt} &= -x + 2y - 2 & \frac{dy}{dt} &= 2x - y - 2 & \text{Eqns 3} \\ & \begin{pmatrix} -1 & 2 \\ 2 & -1 \end{pmatrix} \end{aligned}$$

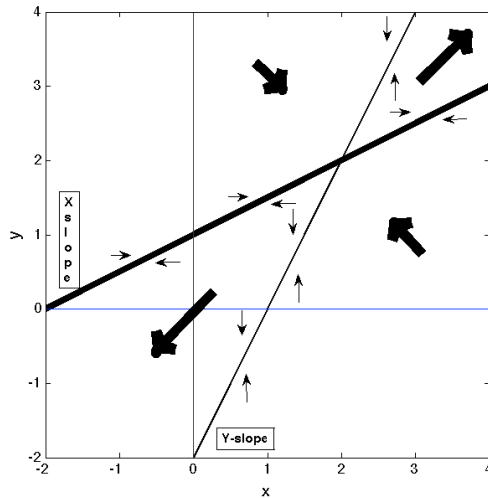


Figure 3: Nullclines for Eqns 3. Thick line ( $dx/dt=0$ ) has now swapped places with thin line ( $dy/dt=0$ ). Response arrows also differ from Figure 2, equilibrium at (2,2) is now unstable.

Here we can see that the equilibrium is unstable. We can note that the connection strengths, the off-diagonal terms in the matrix, also here  $\tan(X\text{-slope})$  and  $\tan(Y\text{-slope})$ , are now 2 rather than 0.5. So anecdotally this conforms to a general picture that larger connection strengths are more conducive to instability; though we should also note that if these connection strengths had been of opposite sign, of whatever strength, stability would have been the consequence. We can now see how this analysis extends to the nonlinear picture.

### May's analysis: nonlinear systems

May (1972) does not lay down any constraints on the very general class of nonlinear systems, bar implicitly that they should be smooth and differentiable so that they can be approximated by a linear system around any equilibrium point under investigation. For simplicity we start by restricting ourselves to systems of the form:

$$\frac{dx_i}{dt} = \sum_j F_{ji}(x_j) \quad \text{Eqns 4}$$

and further restrict the classes of functions to just linear and sigmoid. We can demonstrate our essential points with a two-variable system:

$$\frac{dx}{dt} = -x + y \quad \frac{dy}{dt} = \frac{20}{(1 + e^{(10-x)})} \quad \text{Eqns 5}$$

The numbers have been chosen to demonstrate that there are now several equilibria, as demonstrated by the intersection of nullclines in Figure 4. We can see that two of these equilibria conform to the pattern of Figure 2 (and are stable), whereas the central equilibrium conforms to the pattern of Figure 3 (and is unstable).

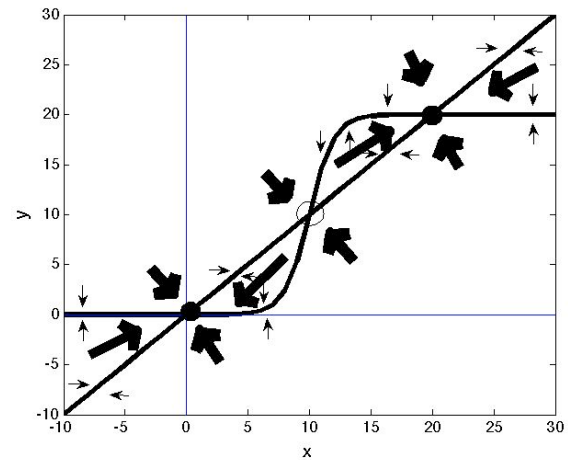


Figure 4: Nullclines for Eqns 5. Three equilibria are circled, the central one (open circle) is unstable, the other two (closed circles) are stable.

### Which Distribution of Connection Strengths?

With the aid of this sketch we can make the trivial observations that the addition of even a single simple monotonic nonlinear function, such as this sigmoid, means that there can be several equilibrium points and that in general the slope of the nonlinear function, related to connection strength, varies from one equilibrium to another.

May wishes to extend the conclusions of the linear analysis – where the probability of stability depends on the nature of the distribution from which connection strengths are drawn – to a nonlinear case with an undefined distribution of nonlinear functions. But this could only be done systematically by firstly specifying the distribution of parameters in the specified class or ensemble of nonlinear functions; and secondly, determining where on such functions one measures the slope. Since there can be several equilibria, this gives several possible values for the connection strength. Given that low connection strengths tend to be conducive to stability in the linear case, it can be noted that many nonlinear functions including these sigmoids have regions where the slope is low.

May would need to do all this to complete his project of generalizing to nonlinear systems. One could then in principle find the distribution of connection strengths over all the equilibria, and perhaps give an estimate of the proportions of these that were stable or unstable. But then further work would need to be done to assess whether the system as a whole was stable or not, since that is a global property.

## Global versus Local Stability

For a system to be globally unstable, every single equilibrium point must be unstable. But for stability it is sufficient for there to be just a single stable equilibrium point within the region of interest. In the case of linear systems, global stability and local stability are one and the same, but May's analysis fails to take account of the fact that nonlinear systems are different. Even if we had an estimate of the probability of any specific equilibrium point being stable, this may well be a gross under-estimate of the chance of there being stability somewhere within the system as a whole.

In some classes of nonlinear functions, e.g. sinusoidal, there is the potential for an unlimited number of intersections with a straight line, corresponding to an unbounded number of equilibria in the two-variable system. For well-behaved curves, as we can see in Figure 4, stable and unstable equilibria alternate so that as long as we have more than one equilibrium we are guaranteed a stable one.

For simplicity, in order to get the main points across, the examples above are restricted to systems of just two variables. Extending this to an  $n$ -variable system with  $n > 2$  requires more analysis. But in summary, the May analysis simply ignores these crucial differences between nonlinear and linear systems, and in doing so typically underestimates, perhaps grossly, the probability of stability in nonlinear systems.

## Ambiguous Circuits

Thomas and colleagues (Thomas and D'Ari, 1990; Thomas and Kaufman, 2001a, 2001b) discuss the roles of positive and negative feedback in nonlinear biological systems. It so happens that their main interest is in the positive feedback circuits that lead to multistationarity, or switching, in genetic regulatory circuits. Nevertheless, much of their analysis can be applied to investigating issues of negative feedback circuits leading to homeostasis or stability. As with May, they are considering a dynamic system of  $n$  variables where many (but typically not all) pairwise interactions are present. This leads to the same connectance or Jacobian matrix. But unlike May they explicitly note that in the general nonlinear case the strengths (and indeed possibly the signs) of these interactions will vary throughout phase space.

Following their analysis, we note that any connectance matrix  $A$  can be considered as composed of multiple overlapping feedback circuits. For any such circuit, the indices are circular permutations of each other. For instance in a 3-variable system as sketched in Figure 5, the full list of potential circuits is:  $\langle a_{11} \rangle$ ,  $\langle a_{22} \rangle$ ,  $\langle a_{33} \rangle$ ,  $\langle a_{12}a_{21} \rangle$ ,  $\langle a_{23}a_{32} \rangle$ ,  $\langle a_{31}a_{13} \rangle$ ,  $\langle a_{12}a_{23}a_{31} \rangle$ ,  $\langle a_{21}a_{13}a_{32} \rangle$ . If one or more of the connections in such a circuit is zero, that circuit as a whole is non-functional; but otherwise, a count-up of the number of negative connection weights decides whether that individual feedback circuit constitutes a negative feedback (odd number of negatives) or positive feedback (even number). The limiting case of such a circuit is that constituted by self-feedback, given by the term  $a_{ii}$  on the main diagonal; that minimal circuit will be non-functional, negative-feedback or positive-feedback depending on whether its value is zero, or its sign is negative or positive.

Thomas and Kaufman (2001a) defined a full-circuit as those circuits and unions of disjoint circuits that involve all the variables of a system. Hence in this 3-variable system, there are six possible full-circuits:

$$\langle a_{11} \cdot a_{22} \cdot a_{33} \rangle, \quad \langle a_{11} \cdot a_{23}a_{32} \rangle, \quad \langle a_{22} \cdot a_{31}a_{13} \rangle, \quad \langle a_{33} \cdot a_{12}a_{21} \rangle, \\ \langle a_{12}a_{23}a_{31} \rangle, \quad \langle a_{13}a_{32}a_{21} \rangle$$

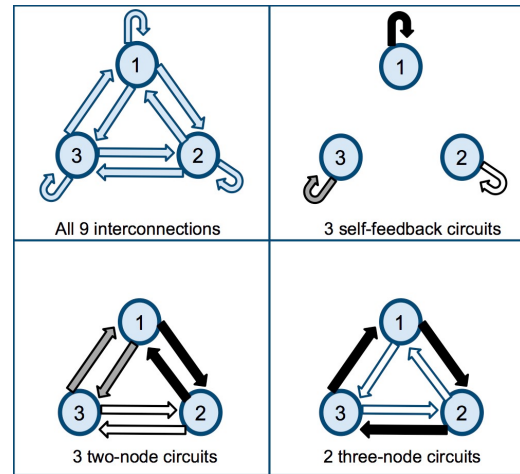


Figure 5: The eight potential circuits, differentiated by shading, within a system of 3 variables fully interconnected.

These correspond to the terms of the determinant of the Jacobian matrix. For any one such full-circuit, considered in isolation, the type of steady state this generates will be determined entirely by the signs, plus or minus, of the various component circuits that comprise this full-circuit. Given that in nonlinear systems any (or all) connection strengths can vary according to position in phase space, and given that the change of sign of any one connection will change the sign of any component circuit of which it is part, we can see that this will alter the type of steady state generated.

This highlights the significance of those connection strengths in a nonlinear system that change in sign as one moves through phase space. These arise from nonmonotonic functions that generate circuits that switch between negative and positive according to context – ‘ambiguous’ circuits – and thereby generate ambiguous full-circuits. Such changes in sign, in one or many such connections, carve up the phase space into different regions, and one can expect the properties of steady states to differ from one such region to the next. This gives the richness of possibilities to nonlinear systems that is missing from the linear ones.

## Plausible nonlinearities

It might be argued that with some systems, although interactions are potentially nonlinear they are ‘linear enough’ for there to be only a single equilibrium. Here we present and discuss some simple nonlinear functions, to see where and how they generate multiple possible equilibria. If one was to analyse fully the probability of stability in some class of nonlinear systems, these might be appropriate simple classes to start on.

## Sigmoids

Sigmoids are commonly used to model physical or biological systems, since they represent an effect that is monotonic yet with asymptotes at lower and upper bounds. A widespread example of where they are used in artificial systems would be Artificial Neural Networks. We have already seen above (Equations 5) that even a simple monotonic function such as a sigmoid is not 'linear enough' to avoid multiple equilibria. The ambiguous circuits discussed above, generating changes in stability through nonmonotonic functions, do not exhaust the ways in which multiple equilibria can exist. Figure 4 demonstrates how both stable and unstable equilibria can be generated merely by a change in strength of a connection without change in sign.

## Piecewise linear with a single bend

Even simpler than a sigmoid, consider a piecewise linear function coupled with a linear function:

$$\frac{dx}{dt} = \max(0, 2 + y - 2x) \quad \frac{dy}{dt} = x - y \quad \text{Eqns 6}$$

These are both linear except that  $dx/dt$  is constrained not to go below zero. As can be seen from Figure 6, this is sufficient to generate a pair of equilibria, one stable and the other unstable.

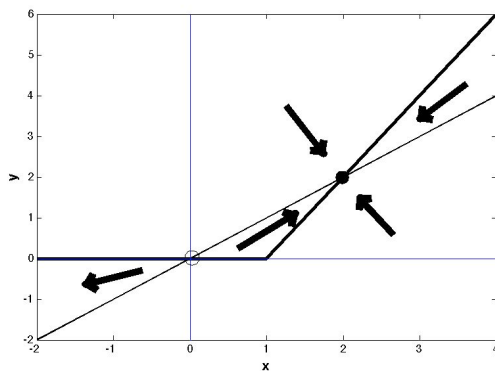


Figure 6: A perturbation analysis of Equations 6, using the same conventions as in Figure 4. There is a stable equilibrium at (2,2) and an unstable equilibrium at the origin (0,0).

## Sinusoidal functions

We have seen how the single inflexion of a sigmoid allows the possibility of 3 intersections with a straight line and hence 3 equilibria. Crudely speaking, the more bends the more possibilities for intersections, and with oscillatory functions such as a sine wave the slope changes in sign repeatedly and indefinitely. The combination of a straight line and a sine wave can lead to an arbitrary number of equilibria that will alternate between stable and unstable. Going further, it can be shown (Kaufman and Thomas, 2002) that a system of 3 variables:

$$\begin{aligned} \frac{dx}{dt} &= -bx + \sin(y) & \frac{dy}{dt} &= -by + \sin(z) \\ \frac{dz}{dt} &= -bz + \sin(x) \end{aligned} \quad \text{Eqns 7}$$

can, depending on the parameter  $b$ , move from having a single steady state for  $b > 1$ , through multiple steady states as  $b$  decreases, with the number of steady states tending to infinity as  $b \rightarrow 0$ . The dynamics change from simple to chaotic, with periodic or multi-periodic windows. The many changes of sign within the regions where nullclines intersect provide ambiguous circuits and increase the richness of possibilities.

## Hat-shaped functions

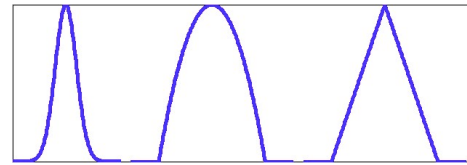


Figure 7: Three 'hat-functions' with broadly similar consequences: gaussian, truncated parabola, and witches hat.

Unimodal 'hat-shaped' functions whose slopes have a single change of sign are an important class of simple nonlinear functions that share some of the asymptotic properties of sigmoids. The examples in Figure 7 share the property of dropping to zero (or approaching zero in the case of a gaussian) each side of a central region. If we take any of these hat-functions as  $y=H(x)$ , this could represent a viability function of an organism or species  $y$  that can only survive (in the case of the gaussian version: survive to any significant level) within some range of values of an environmental variable bounded above and below. These can be considered amongst the most basic of nonmonotonic functions, and it turns out that they do indeed play a crucial role in giving rise to homeostasis, or a particular form of stable equilibrium, in Daisyworld models. Those who use Daisyworld models (which are one class of nonlinear complex system) assert that homeostasis arises naturally in these, whereas many critics such as Kirchner (2002) consider the probability to be vanishingly small unless the parameters are fixed somehow. This controversy illustrates some of the archetypal contrasting viewpoints presented in the complexity-stability debate, and hence we shall review this at greater length.

## Daisyworld

Lovelock introduced the Daisyworld model (Watson and Lovelock, 1983) as a possible explanation of how organisms coupled in mutual feedback with some environmental variable could form a homeostatic system, biotic-environmental, as is proposed in the Gaia Hypothesis (Lovelock, 1972). The Faint young Sun paradox (Sagan and Mullen, 1972) suggests that despite the heat output of the sun changing significantly over the last few billion years the planetary climate has maintained itself around the temperatures conducive for life. The Gaia Hypothesis suggests that this arises through homeostatic properties of the interactions between biota and environment. In the Daisyworld model the organisms (Daisies) have a viability whose dependence on temperature is given by a hat-



function; the truncated parabola version is used in Watson and Lovelock (1983). In turn, through differential absorption or reflection of sunlight, these Black or White Daisies had a positive or negative affect on the same local temperature that influenced their viability. Such systems can be analysed for stability in the context of noise or perturbations at two levels.

In the first instance, any equilibrium state of such a system can be analysed for stability or instability in the presence of small levels of noise; only stable equilibria will persist, and only stable equilibria that have the biota (Daisies) within their viability zone are relevant. But the main interest of Daisyworld models is the extent to which such stable equilibria can persist in the face of major systemic external perturbations, such as major changes in heat output of the sun. It turns out that the Daisyworld temperature is maintained within the viability zone for significantly greater ranges of solar forcing *with* the biotic feedback to the local temperature, as compared to *without* such feedback. This homeostasis arises from the nonmonotonic nature of the hat-function.

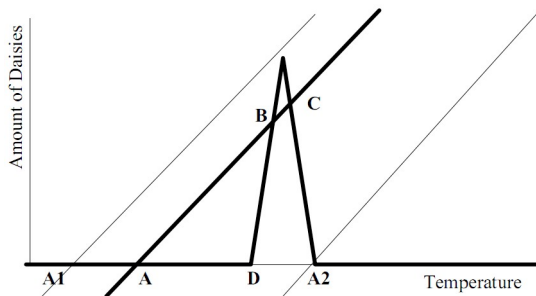


Figure 8: The witches hat-function represents the dependency of Black Daisies on local temperature.

Harvey (2004) showed how a simplification of the Daisyworld model produced the same effects, using a witches hat-function. A reduced version of such homeostasis can be shown with just one species of Daisies, e.g. Black ones. With  $Y$  black daisies, local temperature  $T$ , level of solar forcing  $S$ , then for suitable constants  $k_1$   $k_2$  we have:

$$\frac{dY}{dt} = H(T) - Y \quad \frac{dT}{dt} = S - k_1 T + k_2 Y \quad \text{Eqns 8}$$

The equilibria are shown where the corresponding lines intersect in Figure 8. The different sloping lines, intersecting the temperature axis at A1, A, A2, correspond to different possible levels of solar forcing. It can be seen that, depending on the level of solar forcing, there is either one equilibrium (e.g. at A1 or lower temperatures, or at A2 and higher temperatures) or three (e.g. A, B, C). This latter case gives us: a possible stable equilibrium with zero Daisies at A; or an unstable equilibrium with Daisies at B, the instability being despite the temperature being viable; or a further stable equilibrium at C with Daisies present within their temperature viability-zone. This last stable equilibrium is the focus of interest, and we consider the range of solar forcing for which C exists; i.e., for which there is a stable population of Daisies within the local temperature viability zone. From inspection of Figure 8 we can see that the biotic feedback (from Black

Daisies increasing local temperature) has given rise to viable local temperature over a wider range of solar forcing (corresponding to the range A1↔A2 in the figure) than in the absence of such feedback (corresponding to D↔A2, the unassisted viability range of the hat-function).

Thus the presence of Black Daisies extends the range of viability towards lower solar forcing (the 'faint young sun'); conversely, White Daisies (giving rise to a line ABC with a negative slope in contrast to the positive slope in Figure 8) would extend the range of viability towards higher solar forcing, a hotter sun. This increased range of homeostasis arises from the nonmonotonic nature of the hat-function generating extra possible equilibria.

### Criticism of Daisyworld

This present analysis of the G&A and May papers was originally motivated by work on Daisyworld models (Harvey, 2004) that are one class of these nonlinear systems of a Gaian biota/environment. Such models display homeostasis under a wide range of conditions, yet critics frequently voice the suspicion that this must be because the parameters are carefully chosen from an improbable subset, biased towards negative feedback, in order to achieve stability. For instance Kirchner (2002) suggests that Gaian regulation depends on an implausible assumption that the influence of biota on the environment have a strong tendency to be environment-enhancing rather than environment-degrading. This, it is implied, suggests that such influence has been biased by the modeler to have the appropriate sign, positive or negative. Yet, as is shown in Harvey (2004), regardless of the sign of such a biota→environment effect, when combined with a hat-shaped viability function environment→biota, the resulting ambiguous circuit has the potential for both stable and unstable equilibria within the viability range. Stable equilibria will inevitably be 'selected' in preference to unstable, but since this is independent of the sign of the biota→environment effect it cannot be attributed to some biased choice of this sign. In either case the viable stable equilibrium gives a context that defines this effect as locally environment-enhancing.

This has inevitably been a limited review of the basics of Daisyworld models, missing out many layers of subtlety. For instance the role of hysteresis has not been mentioned, and the significance of those stable equilibria that are within the viability zone, as contrasted with stable equilibria corresponding to extinction, has been treated only briefly. But the main point to be emphasised here is that the interesting (and often counter-intuitive) properties of these models arise from exactly those features of nonlinear systems that May had omitted in his analysis.

Importantly in this context, the homeostasis of Daisyworld systems extends to those with large numbers of variables. Applying these lessons to the construction of artificial systems, it has been demonstrated (Harvey, 2004) that a simulated robot coupled with the environment via an arbitrarily large number of interactions comprising hat-functions (on sensory inputs) and linear functions (on consequent outputs) could find a homeostatic equilibrium. This is so even if the signs of the linear functions are set positively or negatively at random, and the relevant

parameters are varied across some two orders of magnitude. Other examples of systems with multiple interacting component parts achieving equilibrium through the use of hat-functions can be found in Dyke et al. (2007) and McDonald-Gibson et al. (2008). In these cases there was a single environmental variable, and numerous biotic variables subject to hat-function viability limits. The Daisystat (Dyke, 2010) extends this approach to multiple environmental variables.

## Conclusions

The core of this paper is the demonstration that May's (1972) generalization to nonlinear systems – of results that largely hold true in linear systems (Gardner and Ashby, 1970) – is flawed. The method, through linearization around an assumed single equilibrium point, will at best give local stability; there may be many equilibria, and global stability can arise through stability at just one of these. With nonlinear interactions the size, and potentially also the sign, of the connection strengths varies according to position in phase space, and there is no attempt to account for this. To be rigorous, the probability of global stability would depend on assessing the (differing) probabilities of local equilibria, and combining these to calculate the probability that at least one was stable. No attempt at this was presented in (May, 1972), and hence his conclusions should be rejected. His calculations underestimate, potentially by a massive factor, the probability of stability in systems “which in general may obey some quite nonlinear set of first-order differential equations”.

The ambiguous circuits proposed by Thomas and colleagues in their analysis of multistationarity have been used above to explain how a plurality of equilibria can be generated by nonmonotonic functions. But even simple monotonic functions such as a sigmoid can generate alternating stable and unstable equilibria. A number of different simple nonlinear functions were analysed, to demonstrate just how easy it is to breach the assumptions upon which May was relying.

## Does this matter?

Daisyworld models, particularly as the number of variables increase, are just one example of a complex nonlinear system where one would expect May's analysis to be relevant. These demonstrate typical properties of many families of complex nonlinear systems: if one treats the slower variables as parameters and the faster variables as thermal noise, then the remaining variables at intermediate timescales will settle down to a metastable equilibrium (that may be disturbed at a ‘tipping point’ when a ‘parameter’ shifts enough). In ecology it used to be a common view that ecosystems developed through succession towards a single equilibrium state or ‘climax’ (Clements, 1916); but nowadays ecologists are more open to the possible of multiple possible equilibria in an ecosystem.

Our intuitions based on understanding simple linear systems can all too easily lead us into error when considering complex nonlinear ones, with multiple overlapping circuits of feedback. This appears to be the root of the problem here. We are not aware of any previous exposure of these flaws in May (1972); indeed the author is still citing it without qualification

(Haldane and May, 2011) in the context of ‘banking ecosystems’ where clearly there are nonlinearities. It took nearly 30 years for the basic programming errors in G&A to be pointed out in print, and 40 years is too long for these further significant flaws to remain unchallenged.

## Acknowledgments

I thank Nathaniel Virgo and David Waxman for discussions, and for checking that my code was working properly when it failed to replicate G&A. NV found the prior reference to this anomaly (Solow et al., 1999). Anonymous reviewers' comments stimulated the expansion and clarification of some sections; remaining errors and obscurities are the writer's.

## References

- Clements, F. E. (1916). *Plant succession: an analysis of the development of vegetation*. Carnegie Institution of Washington.
- Dyke, J., McDonald-Gibson, J., Di Paolo, E. and Harvey, I. (2007). Increasing Complexity can increase Stability in a Self-regulating Ecosystem. In Almeida e Costa, F., Rocha, L. M., Costa, E., Harvey, I. and Coutinho, A., editors, *Eur. Conf. Artificial Life ECAL 2007*, pages 123–132. LNCS 4648, Springer.
- Dyke, J. (2010). The Daisystat: A model to explore multidimensional homeostasis. In Fellermann, H., Dörr, M., Hanczyc, M., Laursen, L., Maurer, S., Merkle, D., Monnard, P.-A., Stoy K. and Rasmussen S. (Eds.): *Artificial Life XII*. MIT Press.
- Gardner, M. R. and Ashby, W. R. (1970). Connectance of Large Dynamic (Cybernetic) Systems: Critical Values for Stability. *Nature*, 228:784.
- Haldane, A. G. and May, R. M. (2011). Systemic risk in banking ecosystems. *Nature*, 469:351–355.
- Harvey, I. (2004). Homeostasis and Rein Control: From Daisyworld to Active Perception. In Pollack, J., Bedau, M., Husbands, P., Ikegami, T. and Watson, R. A., editors, *Artificial Life IX*, pages 309–314. MIT Press, Cambridge, MA.
- Kaufman, M. and Thomas, R. (2002). Emergence of complex behaviour from simple circuit structures. *C. R. Biologies*, 326:205–214.
- Kirchner, J. W. (2002). The Gaia hypothesis : fact, theory and wishful thinking. *Climatic Change*, 52:391–408.
- Lovelock, J. E. (1972). Gaia as seen through the Atmosphere. *Atmospheric Environment*, 6(8):579–580.
- May, R. M. (1972). Will a Large Complex System be Stable? *Nature*, 238:413–414.
- May, R. M. (1973). *Stability and Complexity in Model Ecosystems*. Princeton University Press, Princeton, New Jersey.
- McDonald-Gibson, J., Dyke, J. G., Di Paolo, E. A. and Harvey, I. R., (2008). Environmental Regulation can arise Under Minimal Assumptions, *J. Th. Biol.*, 251(4):653–666.
- Sagan, C. and Mullen, G. (1972). Earth and Mars : Evolution of Atmospheres and Surface Temperatures, *Science*, 177(4043):52–56.
- Solow, A. R., Costello, C. and Beet, A. (1999). On an Early Result on Stability and Complexity. *The American Naturalist*, 154(5):587–588.
- Thomas, R. and D'Ari, R. (1990). *Biological Feedback*. CRC Press, Boca Raton, Florida, USA.
- Thomas, R. L. and Kaufman, R. (2001a). Multistationarity, the basis of cell differentiation and memory. I. Structural conditions of multistationarity and other nontrivial behavior. *Chaos*, 11(1):170–179.
- Thomas, R. L. and Kaufman, R. (2001b). Multistationarity, the basis of cell differentiation and memory. II. Logical analysis of regulatory networks in terms of feedback circuits. *Chaos*, 11(1):180–195.
- Watson, A. J. and Lovelock, J. E. (1983). Biological Homeostasis of the Global Environment : the Parable of Daisyworld. *Tellus*, 35B:286–289.



# Bondable Cellular Automata

Mark Hatcher<sup>1</sup>, Wolfgang Banzhaf<sup>1</sup> and Tina Yu<sup>1</sup>

<sup>1</sup>Memorial University of Newfoundland, St John's, NL A1B 3X5, Canada

[mhatcher@mun.ca](mailto:mhatcher@mun.ca), [banzhaf@mun.ca](mailto:banzhaf@mun.ca), [tinayu@mun.ca](mailto:tinayu@mun.ca)

## Abstract

We present the Bondable Cellular Automata model, which uses simple 1-dimensional, binary cellular automata as the base atomic elements of an artificial chemistry. Reactions are dependent upon an emergent, 'resolution independent' observable, measurable for individual or composite cellular automata structures. We discuss the rationale behind our choice of observable, 'mean polarity', and behind the choice of a bonding mechanism based on this observable. From simple experimentation we observe that using cellular automata as the underlying dynamical system coupled with mean polarity as the reaction success criterion shows potential to support sustainable emergent behaviour.

## Introduction

The general model for an artificial chemistry consists of the {S (material), R (reaction rule set), A (algorithm)} triplet, with R applied to S according to A (Dittrich et al, 2001). Typically R is hand-coded and applies to a single level of structural hierarchy, while S is composed of atomic types with little or no internal dynamic. This provides ease of analysis at the expense of flexibility.

This work applies a recent, alternate approach of 'sub-symbolic' artificial chemistry, described in RBN-World (Faulconbridge et al, 2009), and in (Faulconbridge et al, 2010), where reactions can apply at any level, with reaction success based upon an internal dynamic of colliding bodies.

Here cellular automata (CA) are used to provide the internal dynamic. By using CA as base atoms it is possible to construct 'Bondable CA' (BCA) systems where the application of R is dependent upon an emergent, possibly 'resolution independent' observable of individual (atomic) or composite (molecular) CA structures. The bonded CA within composite structures are able to exchange state information, introducing new dynamics to the CA, and potentially leading to emergent behaviour and structure in the chemical system.

## Why use Internal Dynamics?

If we allow each body (be it an atom or composite molecule) in an artificial chemistry to possess an internal dynamical system, then we can allow the reactions which occur between bodies to affect the configuration of their systems. Moreover we can allow the reverse: for the configuration of the bodies' systems to affect their ability to react. This way a feedback loop is formed.

This can be achieved by forming reaction rules that are based upon the value of an observable of each body's internal dynamical system. If the chosen observable is measurable for any body of any internal structure or size, be it a single atom or complex molecule, then potentially any two bodies can react with each other, even if their size and structure differ. This allows composite bodies of arbitrary size and structure to be constructed, and allows their dynamical systems to couple and interact. As they grow these composite bodies will take the form of an increasing hierarchy of systems within systems within systems, all interacting with each other.

Further, in the BCA model we have chosen to use an observable that reflects change in the configuration of a body's dynamical system as it occurs, whether or not this change has been caused by reaction with another body. So two bodies that meet the criterion to react with each other at one instance in time might not do so at another instance because of interim change in the values of each body's observable; and vice versa. Similarly, when two bodies react to form a larger, single body, the interaction between their dynamical systems will cause changes to each over time, and might lead to structural instability. If, according to the values of their observables, subcomponents of the single body no longer meet the criterion to remain bonded then decomposition of the single body will occur. Such decomposition will have a knock-on effect upon the internal dynamic of the remaining body, which in turn may cause further, future decomposition, and so on. Thus we have introduced an element of spontaneity to the reactions that take place, allowing them to occur well after or even in the absence of collision between bodies.

So using bodies that possess internal dynamical systems and basing reaction rules upon a suitable observable of those systems allow a rich set of reaction types to take place, between bodies unbounded by size, thus providing a sophisticated platform upon which we can model and explore multi-layered dynamical systems and how they interact.

## Why use Cellular Automata?

The Cellular Automaton is an ideal underlying system for a sub-symbolic chemistry. A tenet of artificial life research is that complex, interesting behaviour may arise from a deceptively simple mechanism, and the cellular automaton is a deceptively simple dynamical system; deceptive because intuition would suggest that from simple rules must emerge simple outcomes. Yet we knew in 1966 that CA with large, intricately constructed transition rules were capable of

universal computation (von Neumann, 1966), and we know now that 1-dimensional CA are capable of performing universal computation (Cook, 2004), or of generating pseudo-random numbers that pass all current tests for randomness (Wolfram, 2002). It is even speculated that CA-type processes are at work in nature, such as the colour patterns generated in the spiral of the Cone Snail, or in the formation of snowflakes (Wolfram, 2002).

Further, a strand of research that allows the CA's transition rule to be altered during iteration has allowed computations to be performed that are not possible with 'simple', standard 1-dimensional CA (Mitchell et al, 1993, 1997), (Kanoh and Wu, 2003), as opposed to those CA which require a partition of cells to store the 'program' to be executed, such as (Cook, 2004). Other work has explored the capability of 2-dimensional CA to perform computations, including universal computation (Sapin et al, 2007).

Thus simple CA, when allowed to interact, have the potential to produce a wider range of behaviours than in isolation. Placing them within the framework of an artificial chemistry, with the continual 'composition, decomposition, re-composition' processes of combinations of CA, allows this potential to be explored in a more open-ended, emergent manner. Figure 1 shows how linking two circular, 1-dimensional CA between just one cell of each can lead to large changes in configuration within a short time, and in automaton B's case interferes with its otherwise short and simple cycle of just 8 configurations.

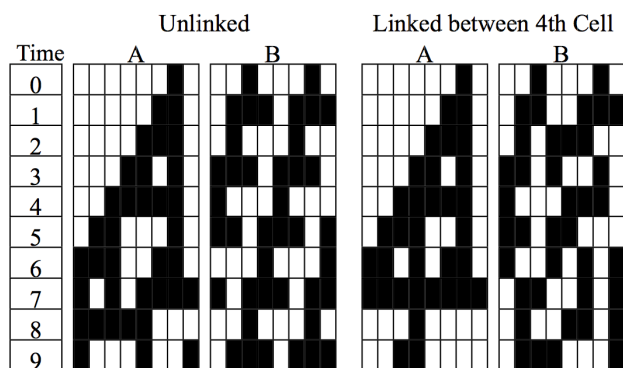


Figure 1: The impact of linking two circular, 1-dimensional cellular automata between one cell of each. The linked cells see each other's neighbourhood when updating. Note that the simple, cyclic configuration of unlinked automaton B becomes disturbed through interaction, and that both CAs' configurations are affected.

This paper describes the BCA model from different perspectives: from the perspective of the individual CA cells, from the perspective of the CA, the *atom*, and from the perspective of the *molecules*, composed of many CA. It describes the *reaction mechanism* that allows composition, or *bonding*, and decomposition, or *unbonding*, to occur between *bodies*, be they atoms or molecules. It provides an example of collision leading to bonding then subsequent, spontaneous decomposition. It describes and discusses the reasoning

behind and the impact of using *mean polarity* as the underlying observable upon which reactions are based. Finally conclusions are drawn, which will steer the direction of future work.

## The BCA Model

The Bondable Cellular Automata model is an artificial chemistry that uses 1-dimensional, binary CA for its base (atomic) elements. These atoms bond to form molecules and molecules further bond to form larger molecules of arbitrary size. Adopting the approach in (Faulconbridge et al, 2009), the reaction rules between bodies (whether they be singular atoms or composite molecules) are not explicitly defined for each type of body, but instead reaction success is based upon the comparison of the value of a single observable for each body; an observable that is based upon the internal CA configurations yet can be measured for any constructible body.

### Model Perspectives

Since BCA is an artificial chemistry based upon cellular automata it is useful to describe and observe it from different perspectives.

**Sub-atomic Level.** BCA can be viewed as a collection of interacting cells. Each cell updates its binary state each iteration, according to the collective state of its perceived neighbourhood of other cells and its assigned transition rule. Figure 2 illustrates the cell's perspective.

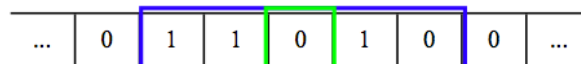


Figure 2: The central (green) cell perceives its neighbourhood as its left and right neighbours, which for rule-width 2 is the 2 cells either side (in blue), but BCA allows a cell to perceive the neighbourhood of a cell in another CA, allowing information to flow between them.

**Atomic Level.** BCA can be viewed as a collection of bondable atoms. Each atom is a circular CA and bonding causes cells in one atom to link to cells in another atom, as shown in Figure 3. Atoms with positive polarity can bond to atoms with negative polarity while atoms with the same polarity cannot bond and atoms with zero polarity are always inert; this is the 'bonding criterion'. At any instant in time an atom's *polarity* is defined as:

$$\text{polarity} = \text{count}(\text{cells in state '1'}) - \text{count}(\text{cells in state '0'})$$

So, in Figure 3 the upper atom has 5 cells in state '1' and 7 cells in state '0', giving it a polarity of  $5 - 7 = -2$  (polarity sign: *negative*). Similarly the lower atom has polarity of  $12 - 0 = +12$  (polarity sign: *positive*). Hence these atoms can bond.

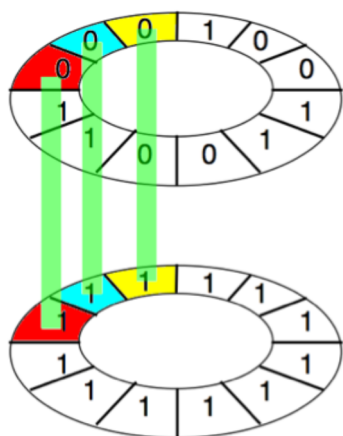


Figure 3: Two bonded atoms are linked between pairs of cells (colour-coded accordingly). Each cell sees its partner's neighbourhood when updating.

Bonding causes cells in one atom to link to cells in the other atoms. Cells can have only one link at a time. Linking causes a cell to view the corresponding cell's neighbourhood rather than its own when updating. Atoms can be bonded to many other atoms, limited only by the availability of unlinked cells. This leads to a rich, complete-graph structure for bonded atoms, as illustrated in Figure 4.

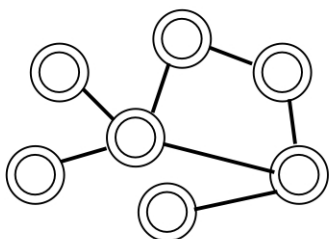


Figure 4: The underlying atomic structure of a molecule. Each atom is a circular CA, and each CA has its own transition rule.

**Molecular Level.** BCA can be viewed as an artificial chemistry, a collection of molecules that collide in pairs and bond.

Due to two body reactions each molecule consists of exactly two (sub) molecules or (conceptually) of a single atom. Figure 5 shows how this leads to a binary-tree structure for molecules, with each parent molecule containing two child molecules.

When molecules collide, if they meet the bonding criterion, then they bond. If they bond this may cause changes to the internal configuration of the CA, which in turn affects polarity, which may lead to sub-components unbonding, which is explained later in the Bonding Example section.

Molecules bond by linking pairs of atoms, one in each pair from each molecule, and how these pairs are chosen and linked is described in the next section. A key aspect of BCA is that the rich underlying atomic structure of molecules is hidden at this molecular level. This greatly simplifies the

description of molecular types and reactions. Also it is possible for two molecules with the same molecular identity to have different structure at the atomic level, thus allowing isomers to be modeled. Moreover, since a molecule's polarity is the emergent outcome of initial configuration, the transition rules and the topology of the underlying atomic bonds, it is possible for two molecules of the same type to possess opposing polarities and thus bond.

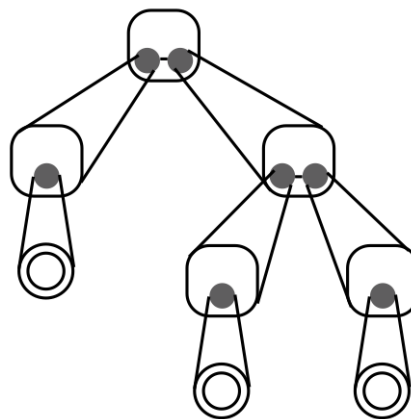


Figure 5: Molecular structure is nested, forming a binary tree. Key: Molecule (rounded square), Atom (outlined ring).

## Material and Reactions

### Atoms and Molecules

In BCA, 1-dimensional, unbounded (circular) binary CA form the atomic elements, their type identified by their transition rule. Atoms can bond to form molecules, and molecules can bond to atoms or other molecules to form ever-larger molecules.

Single atoms behave like standard CA, updating the state of each cell each iteration according to the application of the transition rule to the cell's neighbourhood.

When two atoms are bonded, pairs of cells between each atom are 'linked' together. When updating, linked cells still use their own transition rule but see their partner's neighbourhood instead of their own. Thus the cell states (the 'configuration') of paired atoms affect each other.

### Collisions

BCA's topology is a well-mixed soup, consisting of *bodies*, which can be single (unbonded) atoms or composite molecules. At any point in time collision can randomly occur between any two bodies. Two bodies that collide are known as the reactants, and are tested to see if they will bond according to the bonding criterion.

### Bonding Criterion

Reactants bond according to the mean polarity of each reactant's configuration.

Reactants with opposing polarity signs may bond (positive-to-negative or negative-to-positive) whilst any reactant with

neutral-signed polarity is considered inert and will not react with any other body.

We note that polarity is a scalable, ‘resolution independent’ observable in our model, since it can be measured for single atoms and for molecules of arbitrary size. Polarity is also an emergent property of the configuration of a body, which itself is an emergent property of the initial configuration and transition rule(s) of the underlying CA, and of the effect of the bonds between them. Hence polarity is an emergent property of the underlying dynamical systems and how they interact.

## Bonding Mechanism

**Atomic Level.** Two atoms with opposing polarities bond by forming links between pairs of cells. Which cells link to form the ‘bond site’ is determined by comparison of each CA’s configuration at the time of collision. In keeping with the concept that ‘opposites attract’, the longest continuous run of currently unlinked state 1 cells is identified in the ‘positive’ atom, and the longest run of currently unlinked state 0 cells is identified in the ‘negative’ atom. The shorter of these runs determines the size of the bond site. Each cell in the shorter run is then linked to a cell in the longer run on a 1-to-1 basis until the shorter run is exhausted. Thus every cell has the potential to contribute to a bond site, and therefore to interact with cells in other CA, while the actual location and size of the bond site is an emergent outcome of the automatas’ current configurations.

**Molecular Level.** When two molecules with opposing polarities collide, they form bonds between pairs of atoms, in the manner described above. The molecules will attempt to form bonds between two pairs of atoms, but in practice might form a bond between only one pair, or even not be able to form any bond at all, as the process below explains.

Each molecule is polled for its atom with most positive polarity, and its atom with most negative polarity. Again, by the principle that opposites attract, the most positive atom in the first molecule is paired with the most negative atom in the second molecule (and respectively for the other pair). These atoms bond together at the atomic level in the manner described in the previous section.

Sometimes the chosen pairs of atoms cannot bond, because one or both atoms has no free, unlinked cells with which to form a bond. In this situation any ‘fully linked’ atom is overlooked and the molecule polled for the next most positive/negative atom as appropriate. This process will continue if necessary until either a bondable pair of atoms is found or no more candidates exist. In the latter situation this will lead to the molecules bonding between just one pair of atoms, or in the extreme case not bonding at all.

Allowing two pairs of atoms to possibly form the bond between molecules maintains consistency with the concept that polarity underpins the bonding mechanism, with the most-oppositely polarised atoms in each molecule being attracted to each other and attempting to bond. The key benefit of allowing more than one pair of atoms to bond between molecules is that it allows a rich graph structure to develop at the atomic level, illustrated in Figure 4. If only one pair of atoms were allowed to bond between molecules then this structure would be restricted to a tree, providing less

opportunity for interaction between the atoms within a molecule.

## Unbonding Mechanism

Unbonding occurs spontaneously at the molecular level. If a body consists of more than a single atom, then every iteration of the system the bonds between the two sub-components that form a body are tested. The test is simple: if the two atoms which are actually bonded no longer attract, then they unbond. Unbonding removes all links between paired cells in each atom, and their CA no longer interact.

This unbonding will weaken the link, and hence interaction, between the two sub-components of the body, and if it was the last bond will lead to separation of the body into smaller bodies. Those bodies will then themselves be subject to potential spontaneous decomposition, and so on.

## Bonding Example

Figure 6 illustrates by example the composition of two colliding molecules, and their subsequent decomposition into two new molecules.

Let A and B be molecules in the BCA system. Further, let A be composed of sub-molecules C and D, since this fact will become useful when describing the decomposition stage.

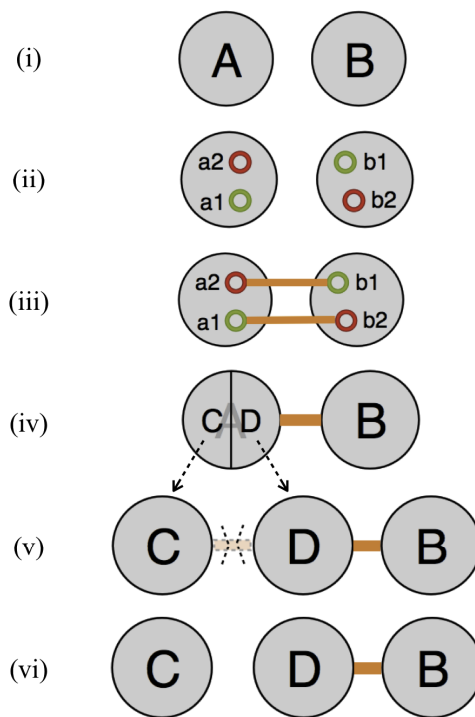
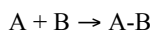


Figure 6: Example of the bonding of two molecules, A and B, and the subsequent, spontaneous decomposition of the resultant into two different molecules, C and D-B.

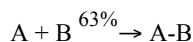
**Composition.** Suppose that A and B collide (Figure 6.i). They have opposing polarities and attempt to bond.

Let the most positive and most negative atoms in A be identified as a1 & a2 (respectively as b1 & b2 in B; see Figure

6.ii). Unlinked cells are available on each atom and so a1 bonds to b2 while a2 bonds to b1 (Figure 6.iii). The molecular equation for this reaction is simply:



A-B denotes composition. Since the reaction success of A + B is not guaranteed for every collision between the types, the current equation is insufficient. So during simulation our system records the percentage reaction success of collision between pairs of types and we can more accurately write the equation as:

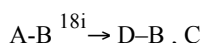


Note that this says nothing about the underlying atomic structure, thus allowing the aforementioned isomers to exist.

**Decomposition.** We now suppose that the formation of A-B leads to changes in the cell states of the underlying CA, through interaction between bonded atoms. These changes subsequently cause the bonded atoms in A's two sub-molecules, C & D (see Figure 6.iv), to no longer attract. Hence the bonds break and A decays (Figure 6.v).

This leads to the breakaway of C as a separate body, while D remains bonded to B, effectively forming a new molecule (Figure 6.vi).

The equation for decomposition is:



$N_i$  represents the number of iterations the composite survived for, and the comma indicates separation. We can consider  $N_i$  as the reaction rate for a decomposing reaction. The full chain of events can be written as:



Some composites will decompose into the original two molecules that formed them, so reversible reactions can be supported by the model.

During the run of a simulation we can track the entire flow of compositions and decompositions for all molecules to derive the reaction network.

## The Impact of Polarity

The choice of polarity as the basis for the bonding criterion followed a process of deduction and experimentation.

Table 1 lists the key, measurable properties of a CA and their suitability for the role. It was quickly identified that any candidate property for underpinning the bonding criterion would need to be not just resolution independent, but would also need to at least in part reflect the dynamic nature of a CA's configuration in order to be an emergent property leading to emergent behaviour.

Therefore rule width, dimension, number of cell states and size were discounted as too trivial to be useful since they remain constant or ignore the CA's configuration; they effectively reduce to static elemental types seen in the general artificial chemistry model. Likewise Transition Rule was

discarded for the above reason and further since it cannot be consistently defined for bonded structures.

	Resolution Independent?	Dynamic?
Rule width	Yes	No
Dimension (1d, 2d, ...)	Yes	No
No. of possible cell states	Yes	No
Size	Yes	Weakly
Transition rule	No	No
Configuration	Yes	Yes

Table 1: Summary of key measurable properties of CA.

This left measures based upon Cell Configuration, which fall into two broad categories: long-term measures and instant measures.

A good example of a long-term measure is cyclelength, as used for RBN-World. We can measure the cyclelength of a CA as the number of iterations required for the configuration to return to a previous state. One strength of using cyclelength is that it is an emergent outcome of bonding; as structures bond the cyclelengths of the sub-components and the combined structure can change. Also, since cyclelength is partly dependent upon other properties of CA such as transition rule and the current configuration, it could provide a valuable reflection of the nature (and specifically Wolfram classes (Wolfram, 1984)) of combined CA.

However one downside to using cyclelength is that its value for a particular body remains static until that body reacts with another body through collision. We believe that using an alternative observable, one whose value can change both because of and independent of reaction with other bodies, adds flexibility to the model since it allows spontaneity and uncertainty to what reactions occur and when.

The other downside to using cyclelength and similar measures is computational cost. Determining the cyclelength of a body in BCA requires direct simulation, since it is dependent upon not just the CA but also how they are bonded; in the worst case its time cost is the Cartesian product of the combined width of a body's CA.

Therefore initially BCA employed the instant measure of polarity at the moment of collision. Instant polarity is an emergent outcome of both internal configuration and the bonding mechanism, is computationally inexpensive (the cost is linear with respect to combined width) and allows for spontaneous unbonding. It also provides underlying consistency to the model: the bonding criterion, the location of bonds in the bonding mechanism (at molecular and atomic levels) and the unbonding mechanism can all be based upon this single characteristic.

Unfortunately during simulation the measure of instant polarity proved to be too stochastic for some transition rules, as Figure 7 demonstrates. The reaction success of colliding bodies became dependent upon the time of collision, which is randomly chosen.

Therefore the measure of mean polarity was implemented. This retains the benefits of instant polarity, including computational cost rising linearly with size, but, as seen in Figure 7, also smoothes the impact of large changes in



configuration between successive iterations. In our experimentation the mean is calculated from the instant that a body is formed, but alternative calculations, such as the mean for the last 100 cycles, could be used, allowing later changes in configuration to have greater impact upon it.

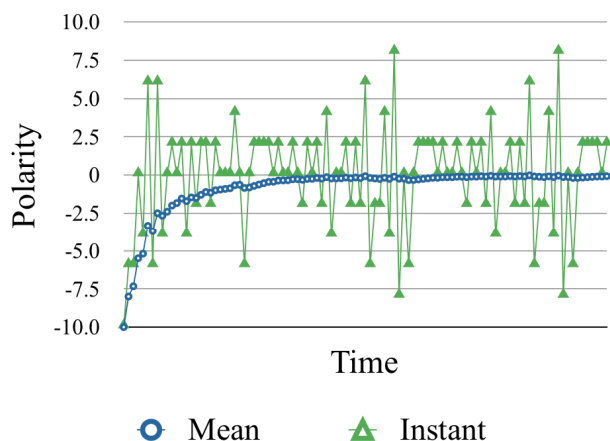


Figure 7: Polarity of a Rule 30 Cellular Automata over time. Whilst the snapshots of polarity change erratically between iterations, mean polarity smoothly settles to a steady value.

## Experiments in Mean Polarity

Rule width	1
Width (number of cells)	12
Dimension	1d
Number of possible cell states	2 (binary)
Number of transition rules	256
Initial configuration of cell states	'000000000001'
<i>Number of iterations:</i>	
Isolated CA	4096
Paired System	100000

Table 2: Set-up for the Isolated and Bonded Pairs experiments.

A key question about mean polarity is whether it would be *too* smooth a measure, essentially reducing in most cases to a static value over time. To answer this question simulations were run using CA with width 12, rule-width 1, providing 256 possible transition rules, or atomic types. Each type of CA was simulated in turn in isolation for 4096 iterations. This is the maximum theoretical cyclelength for a width 12 CA and thus allows a fair calculation of mean polarity over time for the individual CA. Each CA began with the same initial configuration of a single cell set to state '1', all others '0'.

Following this we attempted to bond every possible pair of CA in turn. If they bonded then the simulation was run for a further 10000 iterations, far short of the maximum theoretical 16.7 million iterations required to cover all possible cyclelengths, but in practice sufficient time to determine the long-term sign and magnitude of the pair's mean polarity. Table 2 records the set-up for both experiments. This is the first step into the reaction chemistry of the BCA model.

## Results and Discussion

**Isolated CA.** For 141 out of 256 (56%) of types the sign of mean polarity changes during simulation for isolated atoms. Table 3 shows that although all types begin with negative polarity approximately a quarter of types finish with neutral mean polarity and are hence inert.

Positive	75 (29%)
Negative	115 (45%)
Neutral	66 (26%)

Table 3: Tally of final sign of mean polarity of isolated CA after 4096 iterations.

That such a sizeable proportion of CA types become inert raises concern. However the 'periodic table' of types (Figure 8 in the Appendix) shows that the vast majority of types that achieve neutral mean polarity take more than 256 iterations to do so, and the opportunity for those CA to bond with others and be 'rescued' from inertia remains open during that time.

The periodic table illustrates a strong correlation between the time taken to settle and the amount by which mean polarity changes. All CA begin with mean polarity of -10. Those that finish with that same value never change polarity. Those that finish with mean polarity close to -10 tend to settle within 8 iterations, and as final polarity drifts away from -10, so the time taken to reach the new value tends to increase. We see that many CA types reach high positive polarities, notably acquiring a change of sign, and relatively speaking take their time to do so. This is good since it demonstrates that the mean polarity measure is dynamic in the majority of cases for long enough to present a 'window of opportunity' for different reactions to occur.

## Bonded Pairs.

Total possible unique pairs	32640
Of which bonded	8625 (25%)
<i>Change seen in mean polarity value:</i>	
Individual CA	14049 (81%)
Paired System	7921 (91%)
<i>Change seen in mean polarity sign:</i>	
Individual CA	5799 (34%)
Paired System	3938 (45%)
Both CA and paired system changed polarity	371 (4%)
Both CA changed polarity but paired system did not (the changes 'cancel out')	253 (3%)
One CA's polarity remained stable while the other's and the paired system's changed	402 (5%)

Table 4: Summary Data for the Bonded Pairs experiment.

Table 4 summarises the results of attempting to bond in pairs. The listed percentages for changes seen in polarity are calculated as proportions out of the total number of pairs that succeeded in bonding. We observe that in around 4 out of

every 5 of such cases the act of bonding re-introduces a dynamic element to the value of polarity of the individual CA, and in around 1 out of every 3 cases also leads to a sign change. We also see that in almost half of cases following bonding the paired system's polarity sign changes. This reinforces the view that mean polarity is an emergent outcome of both the underlying CA configurations and the bonding mechanism, not just because it aggregates the values for the subcomponents, but crucially because the subsequent interaction caused by bonding leads to changes in the subcomponents' values themselves.

In the 5% of cases where bonding causes a sign change for one of the CA, if this change were rapid then during a full simulation this would lead to rapid decay of the combined body, releasing both atoms near-instantly back into the soup. This provides the potential for bonding to cause the appearance of unstable molecules, leading to rapid chain reaction; a complement to the slower decay modeled in the unbonding mechanism. In such paired systems, the CA whose sign remains stable is also of interest, since it might have catalytic properties, causing change in the CA it bonds with whilst itself remaining unchanged in polarity sign. Seeking and identifying CA atoms with this property, and possibly even molecules too, is a further step in the research.

Additionally, we observed that bonding causes an overall drift away from neutrality for the bonded pairs, so the act of bonding leaves proportionally fewer inert bodies in the system than if CA were left to iterate in isolation.

These results suggest that rather than locking CA into inert structures with (near) static mean polarity as was feared, bonding could be a self-sustaining process, keeping the system active.

## Conclusion and Future Work

The modeling and polarity experimentation suggest that BCA shows the potential to allow behaviour and structure to be emergent properties of both the bonding mechanism and the underlying CA configurations. More generally it allows us to study the results of the interaction that occurs between simple dynamical systems when they are placed within the framework of an artificial chemistry.

By allowing reactions to occur sometime after or even in absence of a collision BCA is also able to model useful chemical concepts such as variable decay rates, spontaneous reactions derived from internal configuration (rather than due to external trauma), isomers, and catalytic behaviour. Thus we believe that it is a worthy candidate system for the study of emergence.

The experimentation suggests that mean polarity could be an ideal resolution independent observable on which to base reaction success, possibly leading to a positive cycle in system behaviour, where reactions lead to changes in the internal structure of bodies leading to the potential for further reactions. However to assess this further experimentation is needed, including full simulation runs where numerous molecule of many types are present, and are allowed to react to form much larger bodies.

Mean polarity is only one of many possible ways to determine reaction success. Other candidate observables based upon CA configuration exist and can be explored too,

including the possibility of basing reaction success upon a family of measures, or on higher moments. For whichever set of observables we select, we need to strike a balance.

The experimentation using instant polarity has shown that using short-term measures which are based on a computationally inexpensive snapshot of the configuration can lead to essentially stochastic behaviour. Conversely, other long-term measures, such as cyclelength, are less sensitive to short term configuration changes but have increasingly large computational overhead as larger molecules appear, which impinges upon the scalability of experimental simulation. So other candidates will need to be able to display the balance that the use of mean polarity so far achieves, in tempering stochastic influence whilst keeping the computational overhead low.

In the experimentation so far the rate of iteration for the CA has been identical to the rate of iteration for collision. So every time the system performs a collision (or a set of simultaneous collisions) it also updates the configuration of every CA. This need not be so, since we can instead allow the CA to operate in a different time frame and iterate them an arbitrary number of times between each collision.

Further work will examine the reaction networks formed by full simulation of the model, assess the impact of allowing the CA configurations to iterate at a different rate to the collisions and assess the model's viability for application to the modeling of other domains.

## References

- Cook, M. (2004). Universality in Cellular Automata. *Complex Systems*, 15(1):1-40.
- Dittrich, P., Ziegler, J. and Banzhaf, W. (2001). Artificial Chemistries – a Review. *Artificial Life*, 7(3):225-275.
- Faulconbridge, A., Stepney, S., Miller, J. and Caves, L. (2009). RBN-World: Sub-symbolic artificial chemistry. In *Proceedings of the Tenth European Conference on Artificial Life*, 10:371-378.
- Faulconbridge, A., Stepney, S., Miller, J. and Caves, L. (2010). RBN-World: The Hunt for a Rich Artificial chemistry. In *Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*, 12:285-292.
- Kanoh, H. and Wu, Y. (2003). Evolutionary Design of Rule Changing Cellular Automata. *Lecture Notes In Computer Science: Knowledge-Based Intelligent Information and Engineering Systems: 7th International Conference*, 1:258-264.
- Mitchell, M., Hraber, P. and Crutchfield, J. (1993). Revisiting the Edge of Chaos: Evolving Cellular Automata to Perform Computations. *Complex Systems*, 7:89-130.
- Mitchell, M., Hraber, P. and Crutchfield, J. (1997). Evolving Cellular Automata to Perform Computations. *Handbook of Evolutionary Computation*, Oxford University Press.
- von Neumann, J. (1966). *Theory of Self-Reproducing Automata*. University of Illinois, Urbana.
- Sapin, E., Bailleux, O., Chabrier, J-J. and Collet, P. (2007). Demonstration of the Universality of a New Cellular Automaton. *International Journal Of Unconventional Computing*, 3(2):79-104.



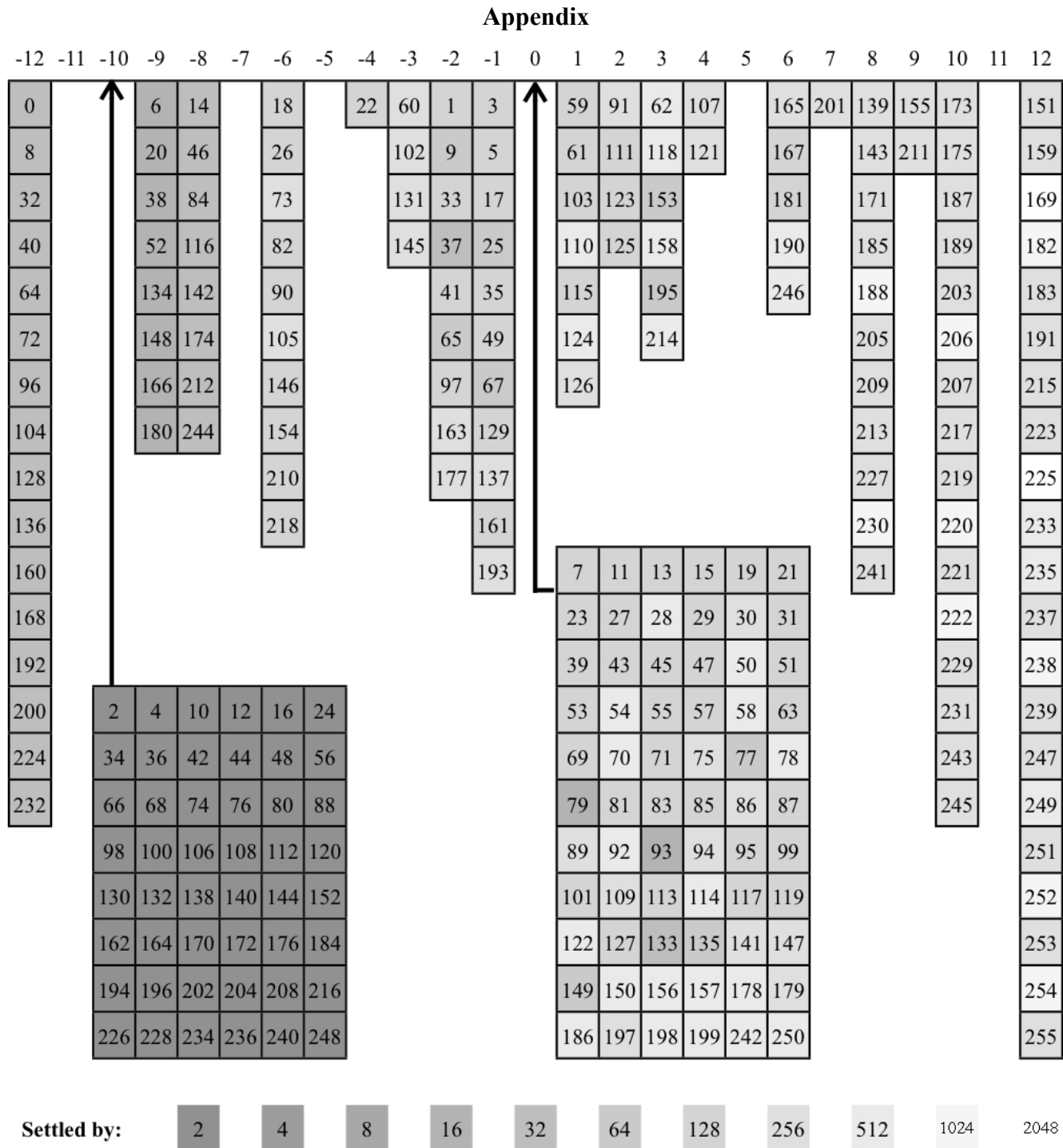


Figure 8. A ‘periodic table’ for the transition rules in BCA. The value at which mean polarity settles for each transition rule in the Isolated CA experiment, with a starting configuration of a single bit set to ‘1’. The scale across the top shows the mean polarity. The number in each box indicates a transition rule. The greyscale shading shows how many iterations it takes for the value of mean polarity to settle to its final value. While individual shades may be hard to discern the general trend is clear.

# Embodied genomes and metaprogramming

Simon Hickinbotham<sup>1</sup>, Susan Stepney<sup>1</sup>, Adam Nellis<sup>1</sup>,  
Tim Clarke<sup>1</sup>, Ed Clark<sup>1</sup>, Mungo Pay<sup>1</sup>, Peter Young<sup>1</sup>

<sup>1</sup>YCCSA, University of York, YO10 5DD, UK

susan@cs.york.ac.uk

## Abstract

We model some of the crucial properties of biological novelty generation, and abstract these out into minimal requirements for an ALife system that exhibits constant novelty generation (open ended evolution) combined with robustness.

The requirements are an embodied genome that supports run-time metaprogramming ('self modifying code'), generation of multiple behaviours expressible as interfaces, and specialisation via (implicit or explicit) removal of interfaces.

The main application of self modifying code to date has been top down, in the branch of Artificial Intelligence concerned with *learning to learn*. However, here we take the bottom up Artificial Life philosophy seriously, and apply the concept to low level behaviours, in order to develop emergent novelty.

## Introduction

It is proving very hard to develop *in silico* ALife systems that exhibit open-ended novelty generation. This may be because many such systems are *closed* in that they often have pre-designed and fixed algorithms, and fixed information representations. The scope for these systems to generate novelty is heavily constrained by these design decisions. This closure is in sharp contrast to biology, where its 'algorithms' and 'representations' are themselves products of the novelty generation processes.

In this paper, we go back to biology, and look at certain aspects of its processes that are key to its power to generate novelty. We use these to develop an open computational novelty generation architecture.

A key source of open-ended biological novelty seems to be the embodiment of the genome in a form that makes it accessible to the other active elements of the system: the DNA can be modified by proteins, changing what future proteins are expressed, and what future modifications occur.

We propose that an analogous approach is needed for open-ended computational innovation. The 'computational DNA' (program code) must be accessible to and modifiable by the active elements (executing program). This can be achieved through *run-time metaprogramming*. (Metaprogramming is when programs manipulate programs; here the

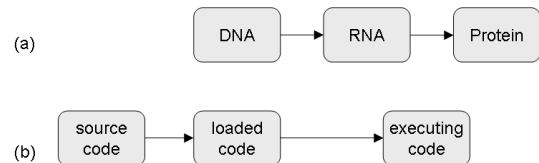


Figure 1: (a) Information flow in the central dogma of molecular biology; (b) control flow in classical computer programs. The vertical alignments indicate rough analogy, discussed later.

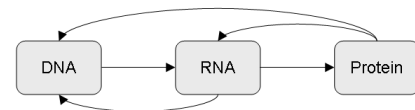


Figure 2: Control flow in the cell

manipulator and manipulated are the *same* program, and the manipulations are performed at run-time. This is also known as reflective programming in high level languages, and as self-modifying code in assembly languages.)

## Biological models

### Self-innovation circular architecture

Crick's central dogma of molecular biology, first stated in 1958 [5], has a linear flow of information content (DNA → RNA → protein, figure 1a). This informational statement is often more strongly interpreted to mean a linear *control pathway*, with DNA 'in control' of the system, and no returning control paths. The standard paradigm of computation has an analogous flow of control (source code → loaded code → executing code). The source code is 'in control'; all subsequent events are a direct consequence of this code (figure 1b).

Such linear flow models are a simple way to describe causality in a system. However, the linear flow of *control* in biology is false. Proteins act on the RNA, and both RNA and proteins act on the DNA, controlling what is expressed, and even changing the DNA (figure 2). There is no strict

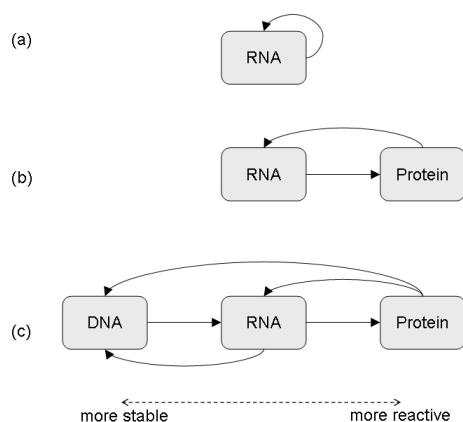


Figure 3: (a) original RNA-world; (b) RNAs and proteins; (c) today's DNA, RNA and protein world. Solid arrows represent the direction of control flow and effect; dashed line represents a molecular stability spectrum.

linear flow of control; it is a closed loop with all entity types able to affect all other entity types. This circularity of interaction allows the emergent biological properties, including novelty generation.

We propose that an analogous circularity of interaction is needed for computational novelty generation: self-modifying self-producing computer *code*, achieved through *run-time metaprogramming*.

## A history of specialisation

Prebiotically there were molecules. Novelty generation resulted in molecules with additional behaviours: RNA encodes information, and can use that information in two ways, as an active *machine*, or as a passive *template*.

In RNA-world [12, 30], the information-bearing template and the active machinery are the same kind of molecules: RNA. However, these two behaviours require different kinds of properties: information-bearing templates require relative stability, whereas the machinery requires reactivity. Biology's solution was to specialise with two sets of molecules: RNA (mainly) for information, and proteins for reactivity. This specialisation continued until today's situation, with the even more stable DNA providing long-term stability for information storage. (See figure 3.)

## The phenotype of the genome

DNA expresses proteins. DNA is composed of *nucleotide bases*; proteins are composed of *amino acids*. These have different reactivity, yet they interact with each other in a variety of ways, such as chemical binding and topological entwining. In particular, different portions of the DNA are physically inaccessible at different stages of the cell cycle. These interactions are subject to selection pressures (limited by physico-chemical constraints), which has led to the

emergence of important biological properties, such as: mutating at differing rates for different genes; specifying when genes express proteins and at what rates; organising the co-location of genes for particular metabolic pathways. These are components of biological innovation.

Analogous properties are not seen to *emerge* in computer simulations (although they can be explicitly designed in).

In order to build a computational analogue of the relevant biological processes, we need to carefully distinguish the genome and the DNA/RNA: (a) the genome is an *abstraction*, a sequence of codes; (b) the DNA (or RNA, in RNA-world) is a physical molecule, *embodying* the genomic information. A protein is another class of physical molecule, its sequence encoded by the genome, physically expressed from the DNA/RNA. In many models of biological evolution the genome and the DNA/RNA that it represents are taken to be synonymous, and the DNA/RNA is modelled differently from the proteins. In reality, however, the genome is an abstraction, and is a different category from the DNA/RNA molecule that is the physical embodiment of that abstraction. The DNA/RNA is an intrinsic part of the *phenotype* of the organism, of the same category as the proteins. Being embodied, it interacts with, and is acted on by, enzymes and metabolites (though less readily than the other entities in the cell).

This embodiment, which we hypothesise is necessary for biological novelty generation, provides the inspiration for our computational architecture to produce analogous open-ended novelty generation *in silico*.

## Computational analogues

We take this aspect of biology, of circular interaction enabled by an embodied template, as inspiration for the design of a computational form of novelty generation.

We perform the following process [27] (for two related biological systems, RNA-world and DNA-world): we produce a model of the *biological* system; we abstract this into a *conceptual* model of the underlying processes and relationships (not shown here); we *instantiate* the conceptual model in computational terms. We use UML class diagrams to express these models.

## AChems as analogues of RNA-world

We first look at the simpler RNA-world (figure 4a). Physics determines how molecules interact, through features such as molecular folding and binding affinities. The genome is an abstraction of the information in the RNA. The biological RNA is embodied: RNA molecules express and are modified by RNA molecules.

The computational analogy is self-modifying code (figure 4b). The analogue of the (disembodied) genome is the (disembodied) source code. The analogue of the active RNA is the executing code: for the analogy to hold, the executing code must be able to modify its own instructions. The

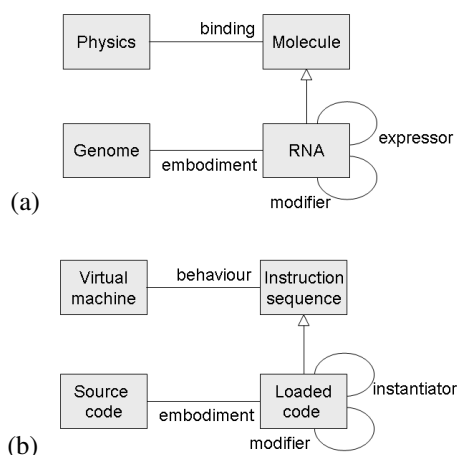


Figure 4: UML class diagram of RNA-world: (a) biological model of embodied RNA; (b) conceptual model instantiated with an AChem

analogue of the physics (which defines how molecules can interact) is the virtual machine (which describes the AChem program language semantics, and how the various AChem objects can interact).

An assembly language level Artificial Chemistry, where the executing code is able to modify the instructions ('embodied source code'), provides a computational model here. Examples include Tierra [26], Avida [1], and stringmol [15, 14, 16], where the 'chemicals' are direct analogues of the RNA strands.

### Reflection as an analogue of DNA-world

Many modern high-level programming languages are designed to enforce a strict separation between code and data, and cannot self-modify in this way. But not all.

We next look at 'DNA-world', a biologically later specialisation of RNA-world (figure 5a). The biological DNA is embodied, and is affected and modified by the proteins it expresses. (Notice this model does not make the biological role of RNA explicit in this process. Here we wish to emphasise the distinction between stable information archive and active machine, so we abstract these as 'DNA' and 'protein' respectively, and omit the intermediate RNA for the purposes of our argument.)

Analogously, the computer source code is *embodied*, and affected and modified by the executing code it specifies (figure 5b). Here we need a programming language where there is a separation between code-representing entities and other active entities (unlike in the RNA/assembly language analogy) that can nevertheless interact at run-time. A high-level language with computational reflection [24] is suitable here: the source code is embodied in the run-time system, and can be modified by the executing system, but is (conceptually) separate from it.

Smalltalk-80 [13] is a good example. In Smalltalk, the

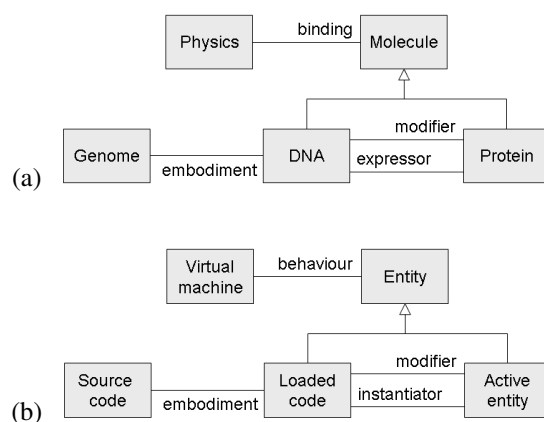


Figure 5: UML class diagram of DNA-world: (a) biological model of embodied DNA and protein machine (omitting the role of RNA, for emphasis); (b) conceptual model instantiated with metaprogramming

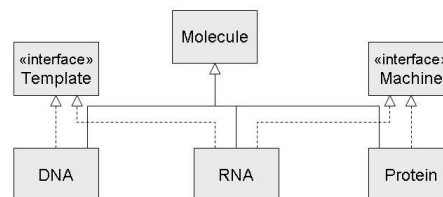


Figure 6: UML class diagram of molecules realising (behavioural) interfaces

source code is just another data structure that can be manipulated by the executing program. Smalltalk is a pure object-oriented language: every value is an object, including classes and code blocks. Code blocks, including ones that modify and create classes, can be constructed at run-time and then executed. An executing Smalltalk system thus has the ability to modify and extend itself: its source code is embodied in the executing system.

Other computationally reflective languages (ones that can modify themselves at run-time, to a greater or lesser extent) include Lisp, Prolog, Python, Ruby, and JavaScript.

### Novelty versus specialisation

RNA encodes information, and can use that information in two ways, as a passive *template*, or expressed as an active *machine*. We can model these two different uses in UML as *interfaces* (figure 6). The interfaces capture the specific behaviours exhibited by certain molecules.

Later (in the RNA-world model), *specialisation* occurred. Molecules that had only one of these behaviours, either machine (protein) or template (DNA), emerged. Once specialised components (components that have *lost* an interface) have emerged, they can adapt to perform their specialisation (remaining interface) more effectively. (Biologically,

specialisation to DNA templates and protein machines was mediated by RNA, and there are still RNA molecules that can be interpreted as fragments of this mediation in modern organisms [3].)

So in modelling terms, novelty generation is *creation* of new interfaces, specialisation is *removal* of interfaces from sub-species of agents. In code terms, removal of an explicit interface is simple: it is just deleted. However, removal of an implicit, emergent interface is not so simple: the low-level behaviour has to change such that the interface behaviour no longer emerges. This is the case both in molecular terms, and in low-level AChem systems. In the molecular case discussed here, this process occurred through differentiation into molecules with distinct chemical structures (nucleotide bases in DNA versus amino acids in proteins). This necessitated the introduction of a *decoding* element to the expression relation, to translate from one structure to the other.

We suggest that such a differentiation step will be helpful in any analogous AChem system designed to progress beyond RNA-world level behaviour in this manner. Specialisation of template and machine behaviours requires templates to be less reactive and machines more reactive. Although such differentiation may be achieved in a homogeneous system (for example, by altering ratios of symbols in the underlying alphabet), it is made easier by having some *structural* difference between them, to help this *behavioural* difference emerge. Given a structural difference, translation will be required to take the template into its machine expression. This translation requirement is not inconsistent with template and machinery being the same kind of thing. There is sufficient richness in chemistry to allow DNA and proteins to be the same kind of thing (molecules) whilst having different representations of their information content (nucleotide bases versus amino acids). The similar form of embodiment allows the information to be modifiable by the system it encodes, whilst the different representations provide the separation of properties that help support specialised behaviour. Chemistry is rich enough to provide this spectrum: AChems will need analogous richness.

High level languages can provide explicit support for this process of specialisation. For example, one pattern supported by refactoring tools is *Extract Interface* [11, p.341]. Aspect oriented programming [18] allows particular kinds of behaviour to cut across the code structure. These are both design time, rather than run time, processes, but some of the concepts may be automatable. Another concept, relevant to implicitly-defined interfaces, is duck typing [20], which allows the type to be determined dynamically, based on what methods a class currently supports.

The ‘softness’ in losing (specialising away) an interface is an important property in terms of robustness through redundancy and degeneracy. It is not necessary for a specialist to lose an interface completely, only for the system to lose reliance in it on providing the interface. The specialist

can safely modify other things about itself, but it might still maintain some ability to implement some part of the interface in an ‘emergency’. If enough parts of the system can implement parts of the interface adequately, then this degeneracy amounts to the system as a whole implementing the whole interface. This provides a form of distributed backup, in case of failure of the machine that is ‘supposed’ to implement the interface.

## Computational architecture

The previous discussion leads us to the notion that, to get emergent novelty in simulation, we should look to run-time metaprogramming. In such a system the code has never finished being written, so the program cannot finish running. Open ended computation is obtained, allowing unprescribed novelty generation within the computer. The main application of self modifying code to date has been top down, in the branch of Artificial Intelligence concerned with *learning to learn* [22, 28, 29]. However, here we take the *bottom up* Artificial Life philosophy seriously, and apply the concept to *low level behaviours*, in order to develop emergent novelty.

Run-time metaprogramming on its own is not sufficient; we also need an architecture within which to run the code. The biological models above can help us here, too. There are two aspects to the architecture. One comes from the class box Physics in figures 4a and 5a, one from the roles modifier and expressor.

## Physics engine

Underlying biology there is physics and chemistry: the processes that define how molecules move around, how they can interact (for example, binding affinities), what the result of the reaction is, and the constraints on the system (for example, conservation laws). In an artificial system, we have to explicitly implement analogues of many of these processes. The usual way to do this is in terms of a *virtual machine* (VM), often referred to as a ‘physics engine’, that provides the execution environment in which the molecule-analogues exist. Tierra [26], for example, has an explicit VM that executes the Tierra assembly language.

The first point to note is that physics is **uncrashable**: there is no real world analogue of a computational core dump or fatal exception. There are two ways to achieve this in the computational architecture: language design or VM handling. The molecular language can be designed such that any molecular interaction results in a legal behaviour. This is relatively straightforward at the assembly language level (care still has to be taken not to access areas outside legal memory). Alternatively, the VM can be designed to trap and isolate any unhandled exceptions. For higher level languages that are modifying themselves, this will become the necessary route.

Next, the VM provides the **spatial dynamics**: how the entities move around, and so who can interact with whom.

This can be explicitly spatial, or be a ‘well mixed’ aspatial model, or even a hybrid (a spatial arrangement of containers with aspatial contents, for example).

We want a system that can generate open-ended novelty without dissolving into chaos. A completely unconstrained system could well modify itself out of existence. Some form of constraint might be needed to allow the system to develop in interesting directions without devolving into a mess of molecule soup. However, a completely constrained system, that allows no modification to its architecture and representations, is static and cannot achieve open-ended dynamics. This is the state of most classic agent-based simulations. The VM should provide such constraint through an **energy model**. This is some analogue of the constraints that real-world physics provides, such as conservation of energy. This provides a limited resource for the various entities; in particular, it prevents ‘free copying’, or unlimited replication, and so provides an evolutionary pressure [8, 17]. It is important not to have a ‘closed’ energy system, however: this would lead to equilibrium. Biological systems are far-from-equilibrium systems, maintained there by an energy *flux*. More sophisticated VMs might also provide an analogue of entropy.

It seems plausible that some degree of constraint between a totally static model, and total freedom, is required; this is possibly some *edge of chaos* [21] requirement. Hence the role of the constraint is to help the system self-organise to maximally complex patterns of structure and behaviour.

Some choices of what goes in the VM and what goes in the molecular language are design decisions. For example, it can be beneficial if the entities have a limited lifetime: this results in entities having to renew themselves to survive, which imposes a natural evolutionary pressure on the system. Whether such a **decay** process is implemented in the VM or in the entities themselves is a design decision: the choice will determine how much the decay can be affected by the intrinsic evolutionary process. The presence of such a decay mechanism has consequences. For example, it means that there will need to be multiple copies of certain machine templates (or templates need to have very different decay properties from active machine molecules), so that the decay of a template does not permanently lose a solution.

## Modification and expression machines

The physics engine provides the VM within which entities can interact and generate novelty (novel entities, novel behaviours, novel interactions). We need some initial entities to set the system going.

Consider the roles modifier and expressor in figures 4a and 5a. In biology, these are embodied, ‘implemented’ by specific machine molecules (ribosomes, transposons, chaperone proteins, etc). Additionally, there are machine molecules that do things not related to self-modification: these are the active molecules performing the external ‘func-

tion’ of the system. This provides a route to embedding application-specific behaviours into a novelty generating architecture.

A novelty generating system could be *bootstrapped* with some specialist machines for these various tasks. this involves writing the bootstraps as code for the embodied templates that, when expressed, becomes the active machine. The key point is that these bootstrap machines *are all encoded on the template*, and so are themselves subject to modification, either directly, by a modifier machine changing their encoding, or through imprecise replication by a ‘sloppy’ replicator machine. And these various modification machines are themselves subject to modification. This is why we are describing only the ‘bootstrap’ architecture: the self-modification processes will then develop new machines, new kinds of machines, and new ways of expressing and otherwise generating machines. This self-modification is what breaks away from fixed algorithms and fixed representations, and allows open-ended novelty generation.

Different kinds of bootstrap machines are suggested by different stages of biological evolution. We could bootstrap with only replicator machines (machines that can copy templates). This is the approach we have taken in our original stringmol AChem [14, 15, 16]. Here we wish to short-circuit the process of evolving all novelty from scratch, but in a way that does not compromise further open-ended novelty generation. We can do so by bootstrapping the system with some more sophisticated machines, some inspired directly by the biological processes of figures 4a and 5a, and some higher level ones implementing ‘non-atomic’ functionality. There is a tension between performance (composing the actions of low level machines versus the single action of a ready-made higher level machine) and flexibility (being able to compose low level machines in novel ways, and having their modifications being more likely to produce viable variant machines). The aim is to engineer a sufficiently powerful and flexible bootstrap that the system can smoothly self-modify into an open ended novelty generator.

Candidate bootstrap machines (which would need to be designed both for the implementation language, and for any application) include those to perform the following functions:

- **expression:** a machine that takes a template, and expresses (instantiates) some machine encoded there. This does not need to be restricted to simple ‘gene expression’: some machines might use information in the template in different way, for example, analogous to the use of ‘gene libraries’ in assembling antibodies. The expression machine might be ‘sloppy’, expressing a range of similar machines, with this sloppiness subject to modification.
- **modification:** a machine that takes a template, and modifies its content in some language dependent way (possibilities include low level machines analogous to transposons

[10], retroviruses [2, 23], and F-plasmids [19], and higher level machines analogous to the processes of gene error correction and crossover, for example).

- **regulation:** a machine that regulates the action of expression machines (this is not explicitly included in the UML models above, but gene regulation is a known critical aspect of biological control, and the regulation is performed by machine-class molecules).
- **replication:** a machine that replicates templates. There will be a constant turnover of templates in RNA-world analogues, and a slower turnover in the more template-stable DNA-world analogues. The replication machine should be ‘sloppy’, providing a source of variation, with this sloppiness subject to modification.
- **translation/transduction:** machines that translate between different information-bearing formats (both internal, and input/output)
- **application:** machines that perform application-specific tasks (the analogue of protein machine behaviours that are not related to modification and expression)

As well as these directly biologically inspired machines, other ‘higher-level’ bootstrap machines might be developed, to help kick-start specific kinds of novelty generation. These are inspired by even later developments in biological evolution. Such machines might include:

- **sensors:** machines that can sense the internal state of the system (for example, via quorum sensing), which information may be used by transducers, regulators, etc
- **generators:** machines that write new templates based on observed behaviours in the system (for example, ‘reverse engineering’ the composed behaviour of several low level machines into a single high-level machine, or breaking down a high level machine into component behaviours)

Other application-specific bootstrap machines can be designed as required. Design of such machines needs to respect the architecture of the system, in particular, the ‘soft’ nature of the mechanisms [4], and the continual turnover of the machines (a good solution, once found, must then be maintained).

Some of these bootstrap machines (particularly higher-level ones) will be easier to implement in high level languages than in assembly-level AChems. However, they are constrained by the particular physics of the system. For example, if the system’s physics does not support global observation, then a global observer machine will not be directly implementable in the system (however, a property akin to global observation could potentially *emerge*). Machines in high level languages can nevertheless be bootstrapped to have potentially sophisticated memories and be-

haviours. There is, however, a tension between the sophistication of the machine that allows it to perform complex functions, and the simplicity of the machine that allows it to be modified in useful ways. Any higher level bootstrap machines should be implemented as compositions of simpler machines wherever possible, allowing modification both of the machines themselves and the ways they are composed. That is, the *representation* of these machines should also be modifiable.

## Biological messiness

Bio-inspired systems are abstractions of the myriad emergent phenomena seen in biology. Their goal is to develop toolsets that *efficiently* distil the unique properties of robustness and adaptability seen in biological systems. Care has to be taken not to throw the baby out with the bathwater, however. We propose that biology generates emergent phenomena by coupling together two phenomena. The first of these is massive *redundancy and degeneracy*, observable in many biological networks: entities are rarely the ‘sole providers’ of all their functionality. This generates massive ‘baseline diversity’. The second is natural selection, which builds hierarchical emergent behaviours by reinforcing beneficial interactions. Crucially *diversity is maintained*, both within and between units of selection, allowing further interactions to be developed and built upon.

This messiness, redundancy and degeneracy that pervades biology has ‘function’, in that it provides a sort of embodied memory. It endows the system with robustness, and alternative pathways should the environment change. It is important not to simplify this away when building abstract models of the processes. In terms of the models introduced above, components should be allowed multiple interfaces, with different components realising different subsets of the complete set of interfaces.

Multiplicity and concentration of machines are an important part of this messiness. Many molecules need to exist in a concentration in order to collectively fulfil their role (DNA being the exception). Given the vast multiplicity of some molecules, ‘erroneous’ molecules that have partial functionality cannot be easily removed, if they do not result in the death of the organism before reproduction. Checking the viability of a molecular unit is an extremely expensive process in biology and is not normally attempted (DNA again being the exception). The continual decay and replenishment is the preferred mechanism. For example, the cell membrane is continually created and consumed [6], and there is a dynamic turnover of flagella motors [7].

This further suggests that there should be multiple copies of templates and machines in the computational system.

## Comparison with existing systems

We are not aware of any high level reflective language systems that fit our DNA-world framework.



A good example of such a computational system that fits our RNA-world framework is an assembly language where the executing code is able to modify the instructions ('embodied source code'). For example, consider an Artificial Chemistry such as Tierra [26], Avida [1], or stringmol [14, 15, 16], which take approaches that are direct analogues of RNA-world. Their chemicals affect and modify each other, by the computational execution of the AChem. However, none fits all the requirements of our framework.

Tierra, directly inspired by RNA-world, fits quite closely with part of our architecture, but has two major differences.

Tierra has an explicit VM to execute its assembly language, designed to be "especially hospitable to synthetic life": non-brittle and evolvable. The spatial model is provided by location in computer memory (although instructions can point to anywhere in space). The entities are analogues of "creatures of the RNA world", although the individual machine instructions are considered to be more analogous to the more chemically active amino acids than to RNA's nucleotide bases. Tierra uses CPU time-slices as an analogue of energy, with the size of the time slice being a tunable function of the entity's size: small size can be rewarded, discouraging 'bloat', or large size can be rewarded, encouraging complexity. It has a decay mechanism in the VM: killing entities when the memory space is close to full. The code can generate errors, which are used to increase the probability of the offending entity being killed. Sloppiness is hardcoded in the VM as bit-flip mutation rates (a background rate, and a higher rate on copy) [9], and through flawed instruction execution. The system is initialised with a single hand-crafted self-replicating entity.

Tierra does not fit our architecture in two important ways.

Firstly, and most importantly, although entities can read and execute the code of other entities, they can modify only themselves (each entity's memory space is write protected). This disallows the emergence of a population of mutually self-modifying entities, other than by copying foreign code into the host entity (a 'pull', rather than a 'push', mechanism). It is a model of single active machines, not of mutually interacting machines mutually defining their properties. This design decision, along with making a less 'brittle' programming language, was made with the aim of overcoming problems in earlier 'Core Wars' implementations (eg, [25]), where mutations mostly just destroyed the system. We believe that the biological inspiration strongly supports mutual modification, however, and that the routes to overcoming the Core Wars issues are a more sophisticated energy model, and a 'softer' language, particularly in respect to binding properties [4].

Secondly, the Tierra energy model is limited. There is no analogue of an energy store (battery, fat reserves) that would enable entities to 'time-shift' their use of the resource, or hand on a surplus to their progeny; Tierra is a 'use it or lose it' model. (Ray [26] mentions a possible extension allowing

capture of CPU slices.) Nevertheless, Tierra evolves an interesting diversity of entities, particularly a range of parasite types.

Avida, although directly inspired by Tierra in the sense that it is an assembly-language based AChem using CPU time slices as a selection pressure, has a very different architecture and motivation from our approach. Entities, in fixed locations in 2D space, interact only with their neighbours, and then only through replication, which copies the replicated entity over its oldest neighbour. Bonus time slices, which can accumulate, are used as an explicit reward mechanism to evolve entities to perform certain tasks.

Stringmol is an assembly language AChem that fits our architecture quite closely, but not perfectly. It is a 'soft' replicator system that has generated novel emergent macro-mutations and hypercycles (two co-dependent species that replicate each other, but are not self-maintaining) [4, 14]. Its execution model involves two strings, and active machine and a passive template; however execution can change either string. The system is initialised with multiple copies of a hand-crafted replication machine, that can replicate any template string it binds to. We have not investigated its behaviour with other kinds of bootstrap machines.

Stringmol has an explicit energy model, in that a certain number of units are added to the container at each timestep, and molecules need to use an amount to execute each instruction. Hence there is a pressure to be small, to enable faster replication cycles. However, the energy is a global resource (energy is not stored in individual entities, but in the system and accessible to all). This removes any incentive for an individual entity to be frugal (beyond replication speed); stringmol exhibits the 'free rider' problem.

## Summary and Conclusions

Biology uses a variety of processes to generate novelty and robustness. Fundamental is the capture of genomic information in an *embodied* genome (DNA or RNA) that is the same kind of structure (molecule) as the active machinery (RNA or proteins). This embodiment allows the active structures to interact with, control, and *modify* the information that defines them. Once novelty has been generated, it can be specialised into different components (DNA as information template, protein as active machine), allowing more effective behaviours to evolve, as the competing requirements of different behaviours are isolated in different components. Specialisation of template and active machinery is aided by different representations (at some level), which require a translation step from information encoded in the template to its expression in the machinery. Specialisation should not go too far: degeneracy and redundancy are also crucial components of biological robustness and adaptability.

Taking these concepts, and abstracting them, we can develop a set of requirements for analogous AChem and ALife implementations: (1) run-time metaprogramming, where

the executing system changes the program that defines its execution, including novelty generation as addition of interfaces; (2) a physics engine VM; (3) specialisation in terms of removal of interfaces (either explicitly, or implicitly by separation of implementation structure); (4) an expression step that decodes information on the template into a different representation on the machine (allowing different kinds of behaviour); (5) redundancy and degeneracy in terms of allowing multiple interfaces per component, and multiple copies of components; (6) sufficiently sophisticated bootstrap machines to short-circuit the origin of life process.

We claim that a suitably ‘rich’ computational environment based on an embodied, modifiable genome that allows novelty generation (adding interfaces) and specialisation (removing interfaces) is a necessary component in maintaining diversity and producing novelty.

## Acknowledgments

This work is part of Plazmid, EPSRC grant EP/F031033/1. Thanks to Alastair Droop and Tim Hoverd for helpful discussions, and to the anonymous referees for their helpful comments.

## References

- [1] C. Adami, C. T. Brown, and W. Kellogg. Evolutionary learning in the 2D artificial life system “Avida”. In *Artificial Life IV*, pages 377–381. MIT Press, 1994.
- [2] J. M. Bishop. Cellular oncogenes and retroviruses. *Ann. Rev. Biochem.*, 52:301–354, 1983.
- [3] T. R. Cech. Exploring the new RNA world, 2004. [http://nobelprize.org/nobel\\_prizes/chemistry/laureates/1989/cech-article.html](http://nobelprize.org/nobel_prizes/chemistry/laureates/1989/cech-article.html).
- [4] E. Clark, A. Nellis, S. Hickinbotham, S. Stepney, T. Clarke, M. Pay, and P. Young. Degeneracy enriches artificial chemistry binding systems. In *ECAL 2011*. MIT Press, 2011.
- [5] F. Crick. Central dogma of molecular biology. *Nature*, 227:561–563, 1970.
- [6] E. A. Dawidowicz. Dynamics of membrane lipid metabolism and turnover. *Ann. Rev. Biochem.*, 56(1):43–57, 1987.
- [7] N. Delalez and J. P. Armitage. Parts exchange: tuning the flagellar motor to fit the conditions. *Mol. Microbiol.*, 71(4):807–10, 2009.
- [8] P. S. di Fenizio. A less abstract artificial chemistry. In *Artificial Life VII*, pages 49–53. MIT Press, 2000.
- [9] A. P. Droop and S. J. Hickinbotham. Application of small-world mutation topologies to an artificial life system. In *ECAL 2011*. MIT Press, 2011.
- [10] C. Feschotte and E. J. Pritham. DNA transposons and the evolution of eukaryotic genomes. *Ann. Rev. Genetics*, 41:331–368, 2007.
- [11] M. Fowler. *Refactoring*. Addison-Wesley, 1999.
- [12] R. F. Gesteland, T. R. Cech, and J. F. Atkins. *The RNA World*. Cold Spring Harbor Press, 3rd edition, 2005.
- [13] A. Goldberg and D. Robson. *Smalltalk-80: The Language and its Implementation*. Addison-Wesley, 1983.
- [14] S. Hickinbotham, E. Clark, S. Stepney, T. Clarke, A. Nellis, M. Pay, and P. Young. Diversity from a monoculture: effects of mutation-on-copy in a string-based artificial chemistry. In *ALife XII*, pages 24–31. MIT Press, 2010.
- [15] S. Hickinbotham, E. Clark, S. Stepney, T. Clarke, A. Nellis, M. Pay, and P. Young. Specification of the stringmol chemical programming language v 0.1. Technical Report YCS-2010-457, University of York, 2010.
- [16] S. Hickinbotham, E. Clark, S. Stepney, T. Clarke, A. Nellis, M. Pay, and P. Young. Molecular microprograms. In *ECAL 2009*, volume 5777 of *LNCS*, pages 291–298. Springer, 2011.
- [17] T. Hoverd and S. Stepney. Energy as a driver of diversity in open-ended evolution. In *ECAL 2011*. MIT Press, 2011.
- [18] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. V. Lopes, J.-M. Loingtier, and J. Irwin. Aspect-oriented programming. In *ECOOP*, volume 1241 of *LNCS*, pages 220–242. Springer, 1997.
- [19] B. C. Kline. A review of mini-F plasmid maintenance. *Plasmid*, 14:1–16, 1983.
- [20] A. Koenig and B. E. Moo. Templates and duck typing. *Dr. Dobbs*, June 2005.
- [21] C. G. Langton. Computation at the edge of chaos: phase transitions and emergent computation. *Physica D*, 42:12–37, 1990.
- [22] D. B. Lenat and J. S. Brown. Why AM and EURISKO appear to work. *Artificial Intelligence*, 23:269–294, 1984.
- [23] R. Lower, J. Lower, and R. Kurth. The viruses in all of us: Characteristics and biological significance of human endogenous retrovirus sequences. *PNAS*, 93:5177–5184, 1996.
- [24] P. Maes. Concepts and experiments in computational reflection. In *OOPSLA’87*, pages 147–155. ACM Press, 1987.
- [25] S. Rasmussen, C. Knudsen, R. Feldberg, and M. Hindsholm. The coreworld: Emergence and evolution of cooperative structures in a computational chemistry. *Physica D*, 42:111–134, 1990.
- [26] T. S. Ray. An Approach to the Synthesis of Life. In *Artificial Life II*, pages 371–408. Addison-Wesley, 1992.
- [27] S. Stepney, R. E. Smith, J. Timmis, A. M. Tyrrell, M. J. Neal, and A. N. W. Hone. Conceptual frameworks for artificial immune systems. *IJUC*, 1(3):315–338, 2005.
- [28] P. Suber. *The Paradox of Self-Amendment: a study of law, logic, omnipotence, and change*. Peter Lang, 1990.
- [29] S. Thrun and L. Y. Pratt, editors. *Learning to Learn*. Kluwer, 1997.
- [30] C. Woese. *The Genetic Code*. Harper & Row, 1968.

# A Gene Regulatory Network Design Of A Synchronous Single-Input Delay Flip-Flop

Imad Hoteit<sup>1</sup>, Nawwaf Kharma<sup>2</sup> and Luc Varin<sup>3</sup>

<sup>1,2</sup>Electrical & Computer Engineering Dept., Concordia University, Montreal (QC), Canada

<sup>3</sup>Biology Department, Concordia University, Montreal (QC), Canada

<sup>1</sup>i\_hoteit@ece.concordia.ca, <sup>2</sup>kharma@ece.concordia.ca, <sup>3</sup>varinl@alcor.concordia.ca

## Abstract

We present a fully detailed design of the very first *synchronous single-input* delay flip-flop (or BioD) implemented as a gene regulatory network in *Escherichia coli* (*E. coli*). The device has one data input (trans-acting RNA), one clock input (far-red light) and an output that reports the state of the device using green fluorescent protein (GFP). The proposed (simulated but not synthesized) device builds on the toggle switch of (Gardner et al., 2000) to provide a more sophisticated device that can be synchronized with other devices in/out of the same cell, and which requires only one input. We provide the first results of a deterministic simulation of a mathematical model of the new device, one which provides evidence that the device is likely to work as required when actually synthesized.

## Introduction

The complex processes that take place in a cell are governed by gene expression which is regulated at several levels during the pathway leading from DNA to protein. Apart from the regulation at the DNA level, gene expression may be regulated during transcription, post-transcription, translation, and during post-translational modification of proteins. Notably, much of the control of gene expression is done either by the regulatory proteins or by mRNAs which are essentially the products of other genes. Hence, the interactions between DNA, RNA, proteins, and other molecules, form a gene regulatory network (GRN). While examining these components individually has provided invaluable information, it is essential (a) to thoroughly investigate these components in variable environments and/or performing variable functions, and (b) to integrate this knowledge to generate valuable genetic devices. Here comes the role of synthetic biology that aims at systematically designing, building, combining and testing new biological functions and systems that do not occur in nature. Indeed individual parts such as promoters and transcription factors can be assembled to synthesize GRNs that perform desired functionalities, such as computing machines.

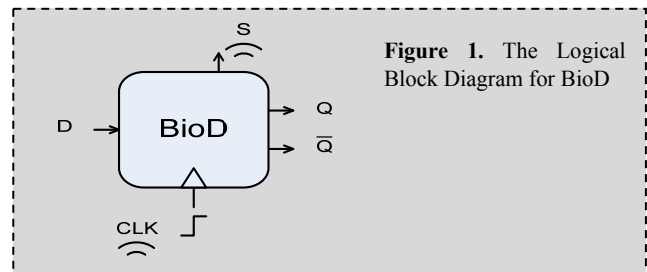
The synthesis of computing machines via the manipulation of DNA (within or without living organisms) started in 1994 when Adleman executed an experimental procedure that used DNA, in vitro, to solve an instance of the directed Hamiltonian path problem (Adleman, 1994). In contrast, in vivo cell-based or cellular computing started in 1998 with the

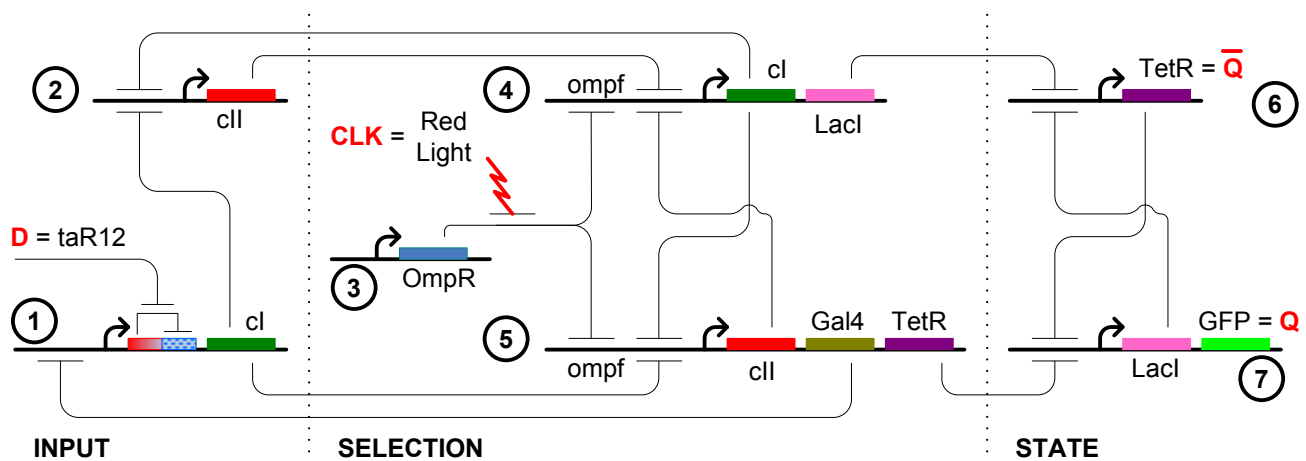
modification of the genome of prokaryotic cells (typically *E. coli*) to realize one- and two-input combinatorial Boolean logic gates (e.g. NOT, AND and IMPLIES) (Knight, Jr. and Sussman, 1998; Weiss et al., 1998); and a similar feat recently was achieved with eukaryotic cells by Kramer (Kramer et al., 2004). Along another dimension, time-dependant or sequential Boolean logic devices have also been implemented in living cells, starting most notably with a 2-input toggle switch by Gardner (Gardner et al., 2000), and a synthetic oscillator by Elowitz (Elowitz and Leibler, 2000). In fact, in one decade this field has grown to generate many elementary devices (Drubin et al., 2007; Boyle and Silver, 2009; Tigges et al., 2009; Haynes and Silver, 2009), including band-pass filters (Stricker et al., 2008) and counters (Friedland et al., 2009). More complicated devices like engineered multicellular pattern generators (Basu et al., 2005), single cell biosensors (Levskaya et al., 2005; Tecon et al., 2006), tumor-targeting bacteria (Anderson et al., 2006), and cell-based computers (Cox, III et al., 2007; Balagadde et al., 2008) have also been built or proposed.

Despite the numerous works on genetic switches, all proposed designs work *asynchronously*. This means that the switch's operation cannot be synchronized with the operation of other parts, using a single global clock. Henceforth, we call a *synchronous single-input delay* switch a *BioD*; a novel GRN that changes states in response to a clock signal by having its output expression follow its input.

## Circuit Design and Modeling

BioD is a synthetic *E. coli* cell that expresses a gene regulatory network acting as a delay switch. By delay switch, we mean a logical device that has an input (*D*), a clock (*CLK*), and an output (*Q*) equal to its state (*S*); see Figure 1 (*Q* is the





**Figure 2.** Gene regulatory network for BioD. The network consists of three sections. *STATE* reflects the state of the network. *SELECTION* affects the *state* switch when the far-red light signal is ON. *INPUT* drives the *selection genes*' activation.

second output and is equal to the logical complement of  $Q$ ). The state of a delay switch is held constant unless and until its input differs from its state, on the rising edge of the clock. In that case, the next state of the delay switch will copy the value of the input (i.e.,  $Q = D$ ). Hence, a cell that acts as a delay switch is effectively a 1-bit memory device, controlled by an input and a clock. The BioD also exhibits its state by expressing (or not) a fluorescent protein.

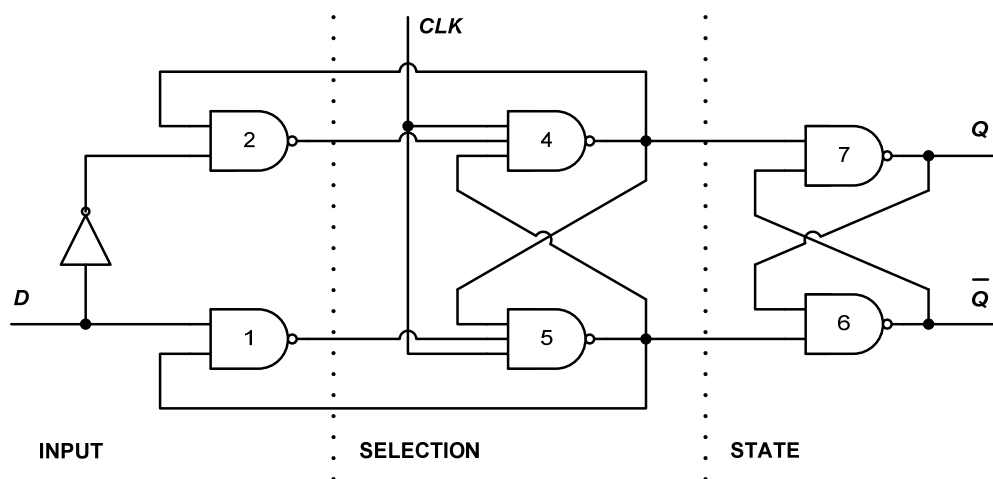
### BioD

BioD has two inputs: *trans-activating* RNA or *taRNA* as input  $D$ , and the presence or absence of far-red light as the clock ( $CLK$ ). It has two complementary outputs ( $Q$  and  $\bar{Q}$ ) defining the state of the flip-flop: the ON state is indicated by the presence of green fluorescence, while the opposite OFF state

is indicated by its absence. As with its electronic counterpart, the output follows the input on the rising edge of the clock. The gene network is comprised of three parts: *input genes*, *state genes* and *selection genes* (as shown in Figure 2).

**Input Genes.** The *input genes* convey to the *selection genes* whether an input signal is present or not. They do so by tipping the balance of the dual-repression of the *selection genes* – discussed below.

In order to sense input  $D$ , gene 1 is designed to be self-repressed, but in such a manner that can only be induced by  $D$ . To achieve this, a form of ribo-regulation is used called *cis*-regulation – which means “acting from the same molecule”. The *cis*-regulation or in our case, *cis*-repression prevents the translation of the gene 1 transcripts by causing them to bend and cover the ribosome binding site (RBS) like a



**Figure 3.** Logic diagram of *BioD* (provided here for simplicity). The above circuit behaves much like the GRN in Figure 2. It is not an exact representation of course, but helps follow the steps the circuit takes to change states. Gene numbers above are matched to gate numbers here. A low  $CLK$  signal neutralizes the selection gates 4 and 5, and sends a high signal (or identity for NAND gates) to the state gates 6 and 7; keeping them unchanged. Since the outputs of input gates 1 and 2 are complements, when the  $CLK$  signal is turned ON, only one of gates 4 or 5 becomes active and thus (i) affects one of the state gates (6 or 7) and (ii) disables its enabling input gate (gate 1 or 2). The input gates are re-enabled after the  $CLK$  goes low, leaving them free to respond to input  $D$ .

lock. The key, comes in the form of trans-activating RNA (taRNA) which, when matched with the *cis*-repressed RNA, unlocks the RBS allowing translation (Isaacs et al., 2004). The taRNA chosen for input *D* is *taR12* which unlocks the *cis*-repression we introduced in gene 1, namely *crR12*.

When the input *D* is present, the transcripts of gene 1 get translated into *cI* proteins (from the  $\lambda$  phage). *cI* in turn represses gene 2. In the absence of input *D* however, the *cis*-repressed transcripts of gene 1 do not get translated into proteins, lifting the repression of gene 2 and allowing its expression.

The presence of input *D* results in the production of the *cI* protein, while its absence results in the production of the *cII* protein (from the P22 phage).

**State Genes.** The *state genes* are very similar to Gardner's toggle switch (Gardner et al., 2000). They consist of two co-repressed genes (i.e. only one expressed at a time), and as such define the state of the BioD device. Genes 7 and 6 represent the complementary outputs *Q* and  $\bar{Q}$  respectively. A green fluorescent protein (GFP) signals the output *Q*, while its absence signals the complementary output  $\bar{Q}$ . The co-repressed nature of the toggle switch means that when either gene is active, it enters into a stable state where it represses the other, and insures its own continued expression. In our case, that stable state can only be affected by the *selection genes*.

As can be seen in Figure 2, the *selection genes* can affect the *state genes* independently of the current state of the BioD. As will be discussed below, genes 4 and 5 are mutually exclusive when active; protecting the *state genes* from conflicting signals. Furthermore, they will either reinforce the repression currently in place in the *state genes* (resulting in no state change), or they will repress the presently dominant gene until the balance is tipped, and the other takes over the state of the device. Which of the two genes 4 or 5 is activated depends on the *input genes* at the time the *CLK* signal is turned ON.

**Selection Genes.** The *selection genes* are always OFF until turned ON by far-red light (the *CLK* input). In the absence of far-red light, genes 4 and 5 are always repressed by the phosphorylated version of OmpR, i.e. OmpRP. Gene 3 is constitutively expressed and produces OmpR. OmpR is phosphorylated in the presence of the EnvZ enzyme. EnvZ is connected to Cph1, which in the presence of far-red light, induces a conformational change in EnvZ preventing the phosphorylation of OmpR. The genes that produce EnvZ and Cph1 (and a few others needed for the light response system (Levskaya et al., 2005)) are not shown in Figure 2.

The phosphorylation of OmpR is dominant in the absence of far-red light and negligible in its presence. Therefore, the far-red light signal causes a drop in OmpRP levels and a corresponding rise in OmpR levels. This drop affects genes 4 and 5 using their promoter, as *ompf* is both activated by OmpR and repressed by OmpRP. Both the functionality of *ompf* and the complementary levels of OmpR and OmpRP result in a system that is quick to start or stop transcription in both genes 4 and 5.

The *selection genes* also respond to and affect the *input genes*. As previously mentioned, BioD is an edge-triggered device, i.e. it responds to the input when the *CLK* signal turns

on, but not to a change in the input when the *CLK* signal is on. This is achieved by designing genes 4 and 5 to only be turned off by the *CLK* signal. When far-red light is introduced, and one of genes 4 or 5 turns on, that gene immediately starts repressing the genes that can potentially repress it; namely, gene 4 represses genes 2 and 5, and gene 5 represses genes 1 and 4. As a result, any change in the input *D* when the *CLK* signal is already on, does not translate to the *selection genes* until the *CLK* signal is turned off, and the repression of the *input genes* is lifted.

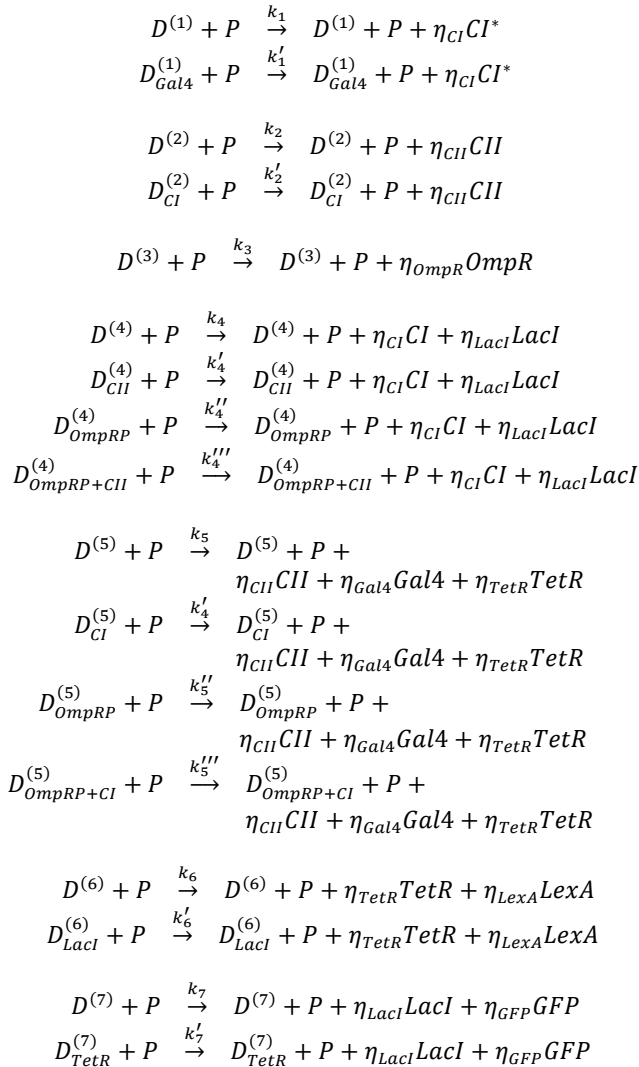
Given that the dynamics of such a gene network are non-trivial, we provide a single fully detailed scenario tracing through one important sequence of transitions. The scenario is that of a change of state, from OFF to ON, in response to a turned ON input (*D*), whose level must stabilize, prior to the introduction of the *CLK* signal (far-red light clocking). When the state of the BioD is OFF, gene 6 is ON, expressing two products. Since one of them (TetR) is repressing gene 7, gene 7 is considered OFF. In the absence of red light, the constitutively expressed (and subsequently phosphorylated) repressor (OmpRP) blocks any production from the *selection genes* (4 and 5). Hence, the status quo of the *state genes* is maintained. Lastly gene 1 is ON, induced by the input (*D*), while gene 2 is OFF, repressed by the product of gene 1, namely *cI*. After clocking, the concentration of OmpRP (which was repressing genes 4 and 5) starts falling. The only other repressor of gene 4 (i.e. *cII* from gene 2) is already OFF. So gene 4 can start producing, and as such, it starts repressing gene 5, which is still repressed by *cI* from gene 1. At this stage, gene 1 is ON, gene 2 is OFF, gene 4 is ON, gene 5 is OFF, while gene 6 is still ON and gene 7 is still OFF. Turning our attention to gene 4, note that one of the repressors it produces is identical to the one generated by gene 7, namely *LacI*. Its production starts switching off gene 6, resulting in a gradual increase in the expression of gene 7. Once gene 7 is fully expressed, it represses gene 6 (via its own *LacI* protein), ensuring the continuation of gene 7's new ON state. Hence, we have achieved a network change of state (indicated by GFP) from OFF to ON (following the value of the input (*D*)). For as long as the *CLK* signal is ON, the new state is maintained. If a significant change in the input level occurs while the clock is ON, the repressions of genes 2 and 5 would not disappear, since gene 4 is ON and produces *cI*. Indeed, as long as gene 4 is ON, it has the ability to keep itself from being repressed by other genes, that is, by repressing them. It is only when the *CLK* signal is removed and both genes 4 and 5 are OFF that the system is free to respond to the input (*D*) again.

### Model

The network in Figure 2 is simulated deterministically. The fast reactions (not shown) involve the binding of proteins to one another and to the DNA. The slow reactions (shown below) involve transcription of mRNA and translation of proteins. The important reactions are presented here as a single combined process.

We define the following terms and chemical species:  $D^{(n)}$ , the DNA protein-binding site in the promoter of gene *n*;  $D_X^{(n)}$ , the  $D^{(n)}$  bound by repressor/activator *X*; *P*, RNA polymerase;  $k_n$ , rate of production of gene *n* (promoter strength);  $k_n^{[1]}$ ,

effective production rate of gene  $n$  after repression/activation;  $\eta_X$ , number of proteins molecules per transcript of gene  $X$ .



where  $CI^*$  depicts the  $cI$  protein that is produced by the *cis*-repressed transcripts of gene 1 – therefore is dependent on input  $D$ .

We simulate the device using a system of rate equations with the concentrations as the dynamical variables. A timing diagram is displayed in Figure 4.

## Results & Discussion

We presented the design of a gene regulatory network (GRN) that, if synthesized and integrated into the genome of an appropriate strain of *E. coli*, will give us a single-cell synchronous single-input toggle switch (we call the *BioD*). The *BioD* accepts as input trans-acting RNA, which allows it be linked to other GRNs; it is clocked using far-red light, which allows external synchronization of its operation; it indicates its state by expressing green fluorescent protein (GFP), which allows easy external monitoring of the state.

The whole design is modular in that it allows alteration of the input sensing and output expression parts without affecting the toggling functionality of the device.

In the sequel, we present the results of simulating the device using a system of rate equations. The results confirm our expectation that the device will toggle when and only when required – though its speed can still be improved.

## Simulation

The core functionality of our *BioD* device is illustrated in Figure 4. The highlighted areas indicate the presence of an input. The reddish hue reflects the presence of the clock input ( $CLK$ ), while the grey diagonal pattern reflects the presence of the data input ( $D$ ). The examples provided have two different data cycles intersecting (or not) with four different clock cycles. This setting allows us to show that the device can indeed go from one state to the other with nothing more than the introduction of the inputs it was designed to respond to; in other words, the device does not get stuck in any one state.

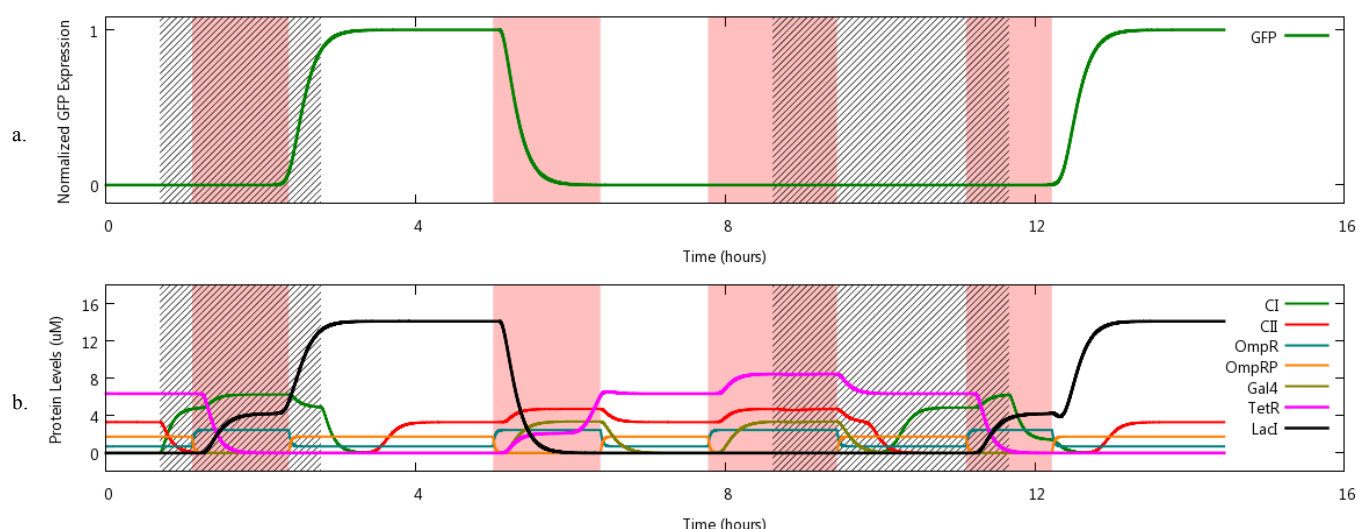
Ideally, with four separate  $CLK$  inputs, the state of the device should follow the  $D$  input four times. In this case, the state should turn ON, then OFF, and then OFF again and finally ON. Figure 4a displays those exact state changes in a deterministic run whose initial condition is an OFF state. The normalized GFP expression output follows the input only at the rising edge of the clock. However while the clock is ON or is OFF, any changes in the input do not propagate to the output. This plot is used to demonstrate the overall input/output relationship. Figure 4b shows the changes in the protein levels – here the levels of  $LexA$  and  $GFP$  were not displayed because they do not affect the behavior of the device.

In the ON level, the expression of a substance is defined mainly by its rates of synthesis and degradation. As expected, some proteins have multiple stable levels of expression. Since  $cI$ ,  $cII$ ,  $LacI$  and  $TetR$  are not only produced in the *selection genes*, but can also be found in either the *input* or *state genes*, the expression of those proteins is significantly increased with the presence of the  $CLK$  signal.  $TetR$  has four levels of expression: (i) OFF, (ii) gene 6 is ON, (iii) gene 5 is ON, and (iv) genes 5 and 6 are ON.  $LacI$  has similar levels of expression using genes 4 and 7. In the case of  $cI$  however, since gene 4 can only turn on if gene 1 is active, it only has three levels of expression (and similarly for  $cII$ ).

Tracing the various signals in Figure 4b shows that, the simulation starts with two active proteins,  $TetR$  (the state of the device is OFF) and  $cII$  (unrepressed since input  $D$  is OFF). Here is a step-by-step explanation of the changes seen in the timing diagram.

First, input  $D$  is introduced, causing the repression of gene 2 (or  $cII$ ) to start. Since gene 1's transcripts are now translated and gene 2 is OFF, gene 4 becomes on an edge-trigger to be turned ON, while gene 5 is doubly repressed by  $OmpRP$  and now by  $cI$ . The  $CLK$  signal is introduced, stopping the phosphorylation of  $OmpR$  and activating gene 4. This raises the levels of  $cI$  and  $LacI$ . The latter represses gene 6 and starts turning the state of the device ON. As  $TetR$  fades away, the GFP levels start climbing. Then the  $CLK$  signal is turned OFF followed by the input  $D$ . These two actions turn OFF gene 4 and disable gene 1 respectively. With both inputs OFF, the  $cI$  repressors produced by genes 1 and 4





**Figure 4.** Deterministic simulation of *BioD*. The two timing diagrams are displaying different signals of the same run. The highlighted areas indicate the presence of an input. The red hue indicates the *CLK* signal (FR light). The grey diagonal pattern indicates the presence of the input *D*. **a.** Normalized GFP expression **b.** Protein levels

degrade without replacement allowing *cII* to return to its previous level. *LacI* which is now produced by gene 7 reaches its un-repressed (ON) state equilibrium.

The second state change occurs when the *CLK* signal is turned on again. Since *cII* is expressed at that time (no input *D*), gene 5 turns ON, causing the repression of gene 1 (through *Gal4*) and the repression of gene 7 (through *TetR*), and raising the level of *cII* (as it is produced by both genes 2 and 5). When the *CLK* is removed, gene 5 is turned OFF, but *cII* and *TetR* remain high, while *Gal4* is repressed. Note that the *TetR* levels are now produced by gene 6 (which took over the state of the toggle switch from gene 7), and no longer by gene 5.

The third *CLK* signal starts now. Gene 5 is again turned ON; the levels of *cII*, *Gal4* and *TetR* climb. In the middle of the *CLK* pulse, the input *D* is introduced. This causes no change in the network. Since input *D* only affects gene 1, its effects were muzzled because the clock had already turned on gene 5 which repressed gene 1. It is only after the clock is turned OFF that the gene 1 repression is lifted. At this point, even though the *CLK* signal is removed, the input *D* is still present, and since gene 1 is no longer repressed by gene 5 (or *Gal4*), *cI* is translated and represses gene 2. The state of the device however does not change since the *state genes* are not directly affected by the *input genes*.

The fourth *CLK* signal turns the state of the device back ON. In the presence of input *D*, the *CLK* turns gene 4 ON causing a similar sequence of events witnessed following the first *CLK* signal.

### A Note Regarding Frequency

The frequency of operation of the *BioD*, that is to say the frequency at which the device can change its state in response to the input is closely related to the genes used to build the network. Indeed, while the design of the *BioD* allows for the use of other genes than the ones presented in this paper,

different genes do have different properties modeled by different synthesis rates, degradation rates, diffusion rates, and promoter/repressor dissociation constants, to name a few. All of these parameters indirectly control the time it takes for the system to respond to an input change, and the time it takes to finish a state change and reach a steady state.

In our case, and when going from an OFF to ON to OFF state, the *CLK* signal had to be sustained for at least 22 minutes to get a sustained state change, while it had to be removed for at least 64 minutes for the network to regain its steady state. That gave the smallest period (or max. frequency) of approximately 86 minutes (5160 seconds).

### A Note Regarding Speed

Speed is a main area of improvement. Indeed, the slowest reactions in a cell are the ones involving repressors and ultimately their transcription and translation. The time it takes to fulfill these operations depends on the promoter strength, the coding sequence that is being transcribed/translated, and the presence of RNA polymerases and/or ribosomes nearby. The impact of repressors is further delayed until the mature protein manages to hit the proper operator site, at the right angle and speed. Using post transcriptional regulation like *taRNA* or RNA interference (RNAi) where possible to effect the state change in *BioD* will make the system significantly faster. The first such place would be where the *selection genes* interact with the *state genes*. Instead of producing repressors for genes 6 or 7, the use of RNAi to prevent one of them from translating repressor proteins would make the entire system significantly faster. Since we already make use of *taR12* to sense the input, we would therefore need another two independent riboregulators that do not interfere with *taR12* or with each other.



## Conclusion & Extension

In this paper, we sought a proof of concept for the first synchronous single-input delay flip-flop implemented as a gene regulatory network in *E. coli*. The simulation we present provides evidence that the device can toggle from the ON state to the OFF state and back, according to the intended functionality. The inherent symmetry of the design reduces the number of genes needed for the device, but introduces some complexity (which is palpable when tracing the various changes the device goes through when toggling).

The BioD is effectively a 1-bit memory element that can operate synchronously (on a clock) with any number of other elements. As such, it can be used to hold the state of a finite state machine, but it could also be used to build a memory bank, an event sequence detector/effector, a decision-making system, and numerous other memory-requiring devices.

## References

- Adleman, L.M. (1994). Molecular computation of solutions to combinatorial problems. *Science* 266, 1021-1024.
- Anderson, J.C., Clarke, E.J., Arkin, A.P., and Voigt, C.A. (2006). Environmentally controlled invasion of cancer cells by engineered bacteria. *J. Mol. Biol.* 355, 619-627.
- Balagadde, F.K., Song, H., Ozaki, J., Collins, C.H., Barnet, M., Arnold, F.H., Quake, S.R., and You, L. (2008). A synthetic *Escherichia coli* predator-prey ecosystem. *Mol. Syst. Biol.* 4, 187.
- Basu, S., Gerchman, Y., Collins, C.H., Arnold, F.H., and Weiss, R. (2005). A synthetic multicellular system for programmed pattern formation. *Nature* 434, 1130-1134.
- Boyle, P.M. and Silver, P.A. (2009). Harnessing nature's toolbox: regulatory elements for synthetic biology. *J. R. Soc. Interface* 6 Suppl 4, S535-S546.
- Cox, R.S., III, Surette, M.G., and Elowitz, M.B. (2007). Programming gene expression with combinatorial promoters. *Mol. Syst. Biol.* 3, 145.
- Drubin, D.A., Way, J.C., and Silver, P.A. (2007). Designing biological systems. *Genes Dev.* 21, 242-254.
- Elowitz, M.B. and Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature* 403, 335-338.
- Friedland, A.E., Lu, T.K., Wang, X., Shi, D., Church, G., and Collins, J.J. (2009). Synthetic gene networks that count. *Science* 324, 1199-1202.
- Gardner, T.S., Cantor, C.R., and Collins, J.J. (2000). Construction of a genetic toggle switch in *Escherichia coli*. *Nature* 403, 339-342.
- Haynes, K.A. and Silver, P.A. (2009). Eukaryotic systems broaden the scope of synthetic biology. *J. Cell Biol.* 187, 589-596.
- Isaacs, F.J., Dwyer, D.J., Ding, C., Pervouchine, D.D., Cantor, C.R., and Collins, J.J. (2004). Engineered riboregulators enable post-transcriptional control of gene expression. *Nat. Biotechnol.* 22, 841-847.
- Knight, T. F., Jr. and Sussman, G. J. Cellular gate technology. UMC98: First International Conference On Unconventional Models Of Computation. [1], 257-272. 1-1-1998.
- Kramer, B.P., Fischer, C., and Fussenegger, M. (2004). BioLogic gates enable logical transcription control in mammalian cells. *Biotechnol. Bioeng.* 87, 478-484.
- Levskaya, A., Chevalier, A.A., Tabor, J.J., Simpson, Z.B., Lavery, L.A., Levy, M., Davidson, E.A., Scouras, A., Ellington, A.D., Marcotte, E.M., and Voigt, C.A. (2005). Synthetic biology: engineering *Escherichia coli* to see light. *Nature* 438, 441-442.
- Stricker, J., Cookson, S., Bennett, M.R., Mather, W.H., Tsimring, L.S., and Hasty, J. (2008). A fast, robust and tunable synthetic gene oscillator. *Nature* 456, 516-519.
- Tecon, R., Wells, M., and van der Meer, J.R. (2006). A new green fluorescent protein-based bacterial biosensor for analysing phenanthrene fluxes. *Environ. Microbiol.* 8, 697-708.
- Tigges, M., Marquez-Lago, T.T., Stelling, J., and Fussenegger, M. (2009). A tunable synthetic mammalian oscillator. *Nature* 457, 309-312.
- Weiss, R., Homsy, G., and Nagpal, R. Programming biological cells. Eighth International Conference on Architectural Support for Programming Languages and Operating Systems. Wild and Crazy Ideas Session [8]. 1998.

# Computational modeling of online reaching

Emmanouil Hourdak<sup>1</sup> and Panos Trahanias<sup>1</sup>

<sup>1</sup>Foundation for Research and Technology  
Institute of Computer Science  
[ehourdak@ics.forth.gr](mailto:ehourdak@ics.forth.gr), [trahania@ics.forth.gr](mailto:trahania@ics.forth.gr)

## Abstract

Humans are able to perform an unlimited repertoire of reaching movements with high accuracy. The skillfulness with which we carry out a given reaching task suggests that there are fundamental control policies that allow us to move our body. In the current paper we examine how an adaptive reach policy can be established, using biologically inspired techniques. The developed model, after an initial imitation phase, can replicate any given trajectory with very good performance.

## Introduction

Reaching is a demanding task, due to the difficulty that lies in the coordination and control of the high-dimensional kinematics of the arm. Despite this fact, primates are able to perform it quite effortlessly. To interface between the symbolic level, that everyday tasks are described, and the low level of motor coordination, the brain uses several intermediate stages of processing (Atkeson, 1989). The richness of human motor abilities suggests that these stages allow the adaptive control of our body, by generalizing motor knowledge to other tasks and directions of movement. Much of this ability to reach is acquired during the early developmental stages of imitation (Piaget, 1962), where infants learn to regulate and control their complex musculoskeletal system.

Reaching motions have widely been studied in order to understand the brain structures that facilitate motor control. Research has revealed that the cerebral cortex uses several different cognitive processes to accomplish this goal, including kinematic (Atkeson, 1989) and dynamic (Soechting and Flanders, 1992) representations of movement, combined with forward and inverse models (Wolpert, 1997). To reduce the complexity of regulating all these processes, the brain makes use of modular structures (Ballard, 1986). Modularity is realized in various levels of the cognitive processing hierarchy and serves to hide the low level spinal system from the higher control centers of the cortex, allowing proper reuse of the motor knowledge.

At the spinal level, converging evidence suggests that modularity is implemented by a pre-coded set of control modules known as primitives (Degallier and Ijspeert, 2010). This concept has received considerable attention in the field of engineering. From a mathematical perspective the method of primitives, or basis functions, is an attractive way to solve the complex nonlinear dynamic equations that are required for motor control. For this reason several models have been proposed, including the VITE model that describes a way to regulate sets of agonist and antagonist muscles to move the limb to a desired state or the FLETE model that consists of a

fixed parameterized system of differential equations that produce basis motor commands (see Degallier and Ijspeert, 2010 for a review). More recent studies in vertebrates suggest a force dependent encoding of motor primitives. For example experiments in paralyzed frogs revealed that limb postures are stored as convergent force fields (Bizzi et al, 1991). In (Giszter et al., 1993) the authors describe how such elementary basis fields can be used to replicate the motor control patterns of a given trajectory.

To be able to reach adaptively, the agent must learn to manipulate its primitives using control policies that generalize across different behaviors. In the cerebral cortex one of the dominant themes used for learning is by receiving rewards from the environment. This paradigm, known as reinforcement learning in engineering, does not require an exact learning signal of the error but rather a scalar, temporally delayed, reward function (Barto, 1995). It is more consistent with the type of feedback provided to humans during learning, where exact information on the error is usually not available. An agent that learns based on reinforcement learning tries to find a policy that will maximize the probability of receiving immediate or future rewards.

In the current paper we investigate how a simulated agent can learn an adaptive reaching policy using methods inspired from biological systems. To accomplish the low level motor control we employ the notion of force fields to design higher order primitives, i.e. motor programs that facilitate the synergetic control of multiple joints. Learning is implemented by modeling the circuitry of the dopaminergic neurons that are responsible for the perception of rewards in the cerebral cortex, and using it to form an adaptive control policy for reaching.

The proposed model consists of several interconnected regions. The roles of these regions are derived based on evidence from imaging and lesion studies that describe their cognitive functions. To implement the modularity at the cortical level we break down the whole system into pathways, i.e. sets of inter-dependent regions that carry out a specific process (Hourdak and Trahanias, 2009). The computational areas in each pathway are modeled using liquid state machines (LSMs, Maass et al, 2002). LSMs consist an alternative to the traditional finite-state machine methods for brain modeling. Their difference lies in that they do not require any convergence to attractor states. Moreover, they are consistent with the homogeneity inherent in the cortical regions where different processing functions are carried out by similar structures (Mountcastle, 1978). To accomplish this dynamic form of processing, LSMs perturb neuronal populations using

continuous or discrete input signals. A large variety of functions can be learned from this perturbation using readout neurons, i.e. neurons implemented with traditionally supervised learning methods. Recently it has been shown that LSMs can carry out any computation with fading memory, provided that the properties of separation and approximation are fulfilled (Mass et al 2002; Hourdakis and Trahanias, 2011).

In the following sections we describe the development and evaluation of a reaching model that is inspired by the aforementioned cortical processes. We begin by examining the biological evidence that underpins the model, and continue to describe the implementation of the model and its evaluation.

## Computational Model

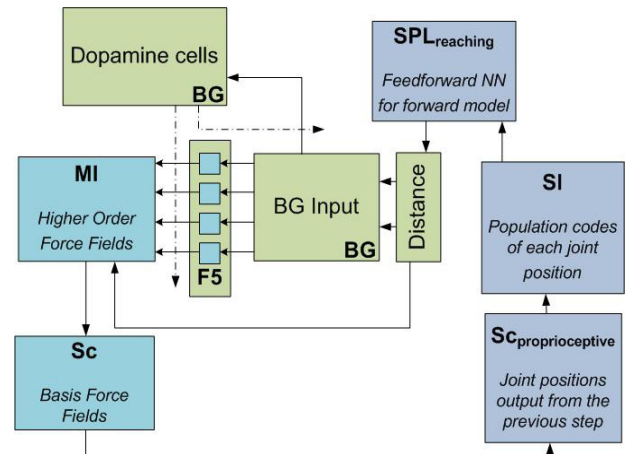
### Cortical model

The central nervous system performs reaching by transforming sequential target locations into muscle commands that move the hand to a desired state (Soechting and Flanders, 1992). To relate the intrinsic proprioceptive state of the agent to extrinsic behavioral goals, such as the points in a trajectory, a forward transformation must be learned (Wolpert, 1997). Anatomical evidence from imaging studies suggest that the cerebral cortex learns such transformations using supervised learning (Doya, 1999). The forward model is implemented in the connections of the primary somatosensory cortex, where the proprioceptive state of the arm is encoded (Sergio and Kalaska, 2003), to the parietal lobe, which is responsible for state estimation. After the behavioral goals have been established, using the forward model and perception, reaching can be accomplished by the adaptive control of primitives. Data from animal lesions and human studies (Sakai et al, 1998) suggest that the basal ganglia are one of the main regions involved in learning sequential movements. This is accomplished by processing the rewards of the environment, which in the brain are evident from the secretion of dopamine, in order to gate motor programs (Thach et al, 2000). Learning of new motor policies is implemented in the projections of the basal ganglia with regions of the prefrontal cortex, where segments of motor acts are encoded (Jeannerod et al., 1995), and primary motor cortex, where the neurons' activity is strongly correlated to the level of activation of individual muscles (Todorov, 2000). Finally, lower level control is mediated by the connections of the primary motor cortex to the spinal cord (Dum and Strick, 1991).

To model the interaction between the sensori and motor systems in the cerebral cortex we use the notion of pathways (Hourdakis and Trahanias, 2009). Each pathway implements a distinct cognitive function and is defined by two factors: (i) the regions that participate in its processing and (ii) the directionality of the information as it progresses the levels of the cognitive hierarchy. This type of abstraction helps to identify and describe at a computational level how cognitive functions are carried out neurally. As a result development of

the model becomes a two stage process; first individual cognitive functions are designed, and then integrated together in order to achieve the required behavioral tasks. The modularity induced by pathways allows us to overcome traditional problems with large scale distributed architectures, such as cross talk. This type of modular approach provides important benefits in computational modeling since it allows identifying how the complex processes that exist in biological systems can be modeled with computational principles.

To design a reaching model, we identify three different pathways: (i) motor control, (ii) reward assignment and (iii) forward model. These are displayed in the following figure, where each pathway is marked with a different color.



**Fig. 1.** The complete layout of our model with the three pathways, motor (blue), reward assignment (green) and forward model (purple).

Motor control (marked in blue) is responsible for the encoding of the primitive model. It includes regions **Sc**, where a set of basis primitives are hardwired, **MI**, where the basis modules are combined into higher order control modules and **F5**, where the higher order control modules are synthesized based on an adaptive reaching control policy. The latter is learned implicitly through the reward assignment pathway (marked in green). Finally the forward transformation of the body-centered state of the agent is accomplished in the forward (marked in blue) pathway. In the proposed model there is also an additional visual perception pathway that handles the perception of the trajectory. However, due to space constraints, in the current paper we assume that the trajectory is given to the agent as a series of consecutive points. In the following sections we describe the mathematical framework that underpins our model, as well as the implementation of each pathway.

### Arm control

To model the effect that the torques have on the joints of the robot we use established laws from control theory (Paul, 1981). The second order kinematics of the robot hand are modeled using the following equation:

$$D(q, \dot{q}, \ddot{q}) = H(q)\ddot{q} + C(q, \dot{q})\dot{q} \quad (1)$$

where  $D$  is the controller that produces the torques that must be applied on the joints of the robot given its state  $q$ , and its first and second order derivatives,  $\dot{q}$  and  $\ddot{q}$  respectively.  $H$  is the joint-space inertia matrix and  $C$  describes the Coriolis and centripetal effects from the joint movement. Eq. 1 can be extended with additional terms such as the viscosity of the joints or the gravity loading of the plant. In the current paper, we applied the model on a simulated frictionless two-link plant, and therefore we didn't include these parameters.

The aim of the computational model is to derive the appropriate local control laws that will allow the plant to reach towards any location. In practice we look for a control policy that will map the state vector of the robot to a control vector from the computational model in a way that minimizes the error of reaching. Degallier and Ijspeert (Degallier and Ijspeert, 2010), suggested that such a control policy  $\pi$  can be defined as:

$$v = \pi(q, t, a) \quad (2)$$

where  $v$  are the joint torques that will be applied to the robot,  $q$  is the state space vector,  $t$  stands for time and  $a$  is the parameterization of the computational model.

The output of our model is the signal produced by the spinal cord circuit. In a biological agent the torques produced would be applied to the hand and result in movement. However since we use a simulated agent we find the second order kinematics of the hand by integrating eq. 1 and solving against the acceleration:

$$\ddot{q} = H(q)^{-1} \{ \tau_p - C(q, \dot{q}) \} \quad (3)$$

The next configuration state of the robot is calculated using the acceleration  $\ddot{q}$  from the equation above, where  $H$ ,  $C$ ,  $q$  and  $\dot{q}$  are as in eq. 1. The goal of the computational model is to produce the appropriate  $\tau_p$  vector of joint torques that will enable the agent to perform reaching.

To evaluate the proposed model we use a simulated two-link planar arm. Control is accomplished by applying torques to the elbow and shoulder joints respectively. Therefore in the presented simulations the  $\tau_p$  vector is two dimensional.

### Forward model pathway

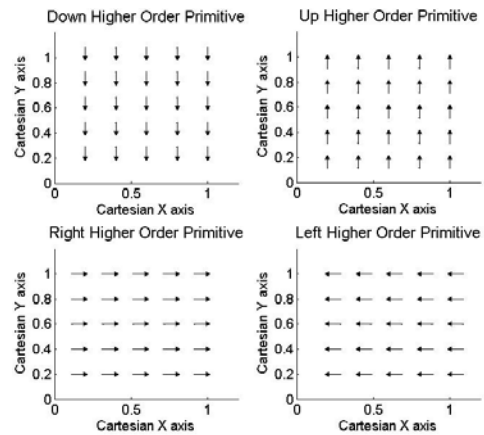
One of the main transformations that takes place during reaching is the cognitive implementation of a forward model (Wolpert, 1997). In the current paper, the forward model is implemented in the regions of the somatosensory and parietal lobe, and allows the agent to approximate the end point position of its hand using the proprioceptive input from the spinal cord.

To accomplish this we have designed the SI network to encode the proprioceptive state of the agent using population codes. This is inspired from the local receptive fields that exist in this region and the somatotopic organization of the SI (Kaas et al., 1979). Population codes assume a fixed tuning profile of the neuron, and therefore can provide a consistent representation of the encoded variable. To learn the forward transformation we train a feedforward neural network in the SPL region that learns to transform the state of the plant to a Cartesian x, y coordinate.

### Motor pathway

Due to the high nonlinearity and dimensionality that is inherent in controlling the arm, devising an appropriate policy for learning to reach can be quite demanding. In the current paper this policy is established upon a few higher order primitives, i.e. self-organized spinal circuits that coordinate elementary motor behaviors. It turns out that, in the adopted planar arm, in order to perform any reaching behavior, only four higher order primitives are required namely up, down, left and right (Fig. 2). In humans such modules are formed during the first stages of the vertebrate motor development.

In order to make the agent generalize motor knowledge to different domains, the primitive model must be consistent with two properties: (i) superposition, i.e. the ability to combine different basis modules together and (ii) invariance, so that it can be scaled appropriately. Primitives based on force fields satisfy these properties (Giszter et al, 1993). As a result by weighting and summing the four higher order primitives shown in Fig. 2 we can produce any motor pattern required.



**Fig. 2.** The higher order primitive model proposed. The four plots show the force map of the primitive, i.e. the forces that are applied to the end position of the limb when the corresponding primitive is active. In the current model we use four different modules, namely up, down, left and right.

The higher order primitives are composed from a set of basis torque fields, implemented in the Sc module. By deriving the force fields using basis torque fields, the primitive model creates a direct mapping between the state space of the robot (i.e. joint values and torques) and the Cartesian space that the trajectory must be planned in (i.e. forces and Cartesian positions), resembling the way motions are processed by humans. We first define each torque field in the workspace of the robot, and then transform it to its corresponding force field. Each torque field is described by a Gaussian multivariate potential function:

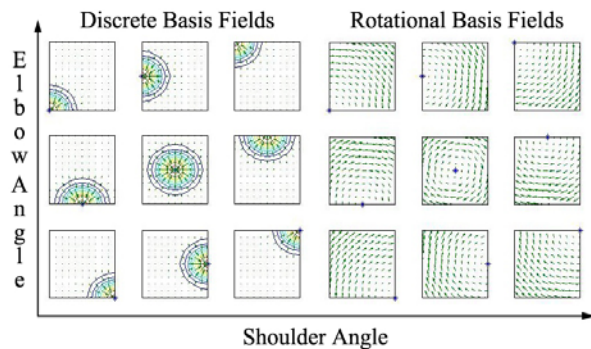
$$G(q, q_0^i) = -e^{\left( \frac{(q - q_0^i)^T K^i (q - q_0^i)}{2} \right)} \quad (4)$$

where  $q_0^i$  is the equilibrium configuration of each torque field,  $q$  is the robot's angle and  $K^i$  a stiffness matrix. The torque

applied by the field is derived using the gradient of the potential function:

$$\tau^i(q) = \nabla G(q, q_0^i) = K^i(q - q_0^i)G(q, q_0^i) \quad (5)$$

Previous research has indicated that in order to achieve stability, two types of primitives must be defined: discrete and rotational (Degallier and Ijspeert, 2010). The rotational primitives are harmonic oscillators associated with a joint. The discrete ones apply a force on the hand based on a shaped valley with different equilibrium points. To ensure good convergence properties we have used 9 discrete and 9 rotational basis torque fields, spread throughout different locations of the robot's workspace (Fig. 3). These are generated from eq. 5 using different stiffness matrices. To generate the discrete torque fields (left block in Fig. 3) we use a semi-definite skew symmetric matrix  $K_{disc}$ , while to generate the rotational fields we use a rotation matrix,  $K_{rot}$ .



**Fig. 3.** Nine basis discrete (left block) and rotational fields (right block) scattered along the  $-\pi.. \pi$  configuration space of the robot. On each subplot the x axis represents the elbow angle of the robot while the y axis represents the shoulder angle. The two stiffness matrices used to generate the fields are  $K_{disc} = \begin{bmatrix} -0.672 & 0 \\ 0 & -0.908 \end{bmatrix}$  and  $K_{rot} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ .

Each plot in Fig. 3 shows the gradient of each torque field. The axes correspond to the  $q_1, q_2$  joint values of the robot's hand. Since we want the model of higher order primitives to be based on the forces that act on the end point of the limb, we need to derive the appropriate torque to force transformation. To accomplish this we convert a torque field to its corresponding force field using the following equation:

$$\varphi = J^T * \tau \quad (6)$$

In eq. 6,  $\tau$  is the torque produced by a torque field while  $\varphi$  is the corresponding force that will be acted to the end point of the plant if the torques are applied.  $J$  is the robot's Jacobian. In the current implementation where the plant is located in a 2 dimensional workspace, the  $6 \times 3$  Jacobian matrix can be constrained to a  $2 \times 2$  matrix as:

$$J = \begin{bmatrix} -l_1 * \sin(q_1) + l_2 * \sin(q_1 + q_2) & -l_2 * \sin(q_1 + q_2) \\ l_1 * \cos(q_1) + l_2 * \cos(q_1 + q_2) & l_2 * \cos(q_1 + q_2) \end{bmatrix} \quad (7)$$

Each higher order force field from Fig. 2 is composed by summing and weighting the basis force fields from eq. 6. To find the weight coefficients, we form a system of  $N$  linear

equations by sampling  $M$  vectors  $P$  from the robot's operational space, for all  $B$  basis force fields.

$$\begin{bmatrix} \varphi_1^1(x^1) & \dots & \varphi_1^B(x^1) \\ \vdots & \ddots & \vdots \\ \varphi_1^1(x^M) & \dots & \varphi_1^B(x^M) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_M \end{bmatrix} = \begin{bmatrix} P_1^1 \\ \vdots \\ P_2^M \end{bmatrix} \quad (8)$$

Each higher order force field is formed by summing and scaling the basis order force fields with the weight coefficients  $a$ . The vector  $a$  is obtained from the least squares solution to the problem:

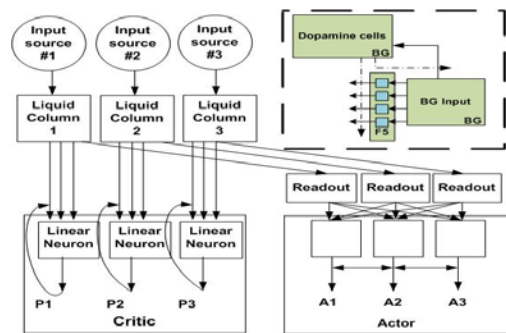
$$\Phi * \alpha = P \quad (9)$$

In the results section we show the force fields that are produced by solving the system in eq. 9, as well as how the plant moves in response to a higher order force field.

## Reward assignment pathway

One of the main methods of primate learning is by obtaining rewards from the environment. In the cerebral cortex, reward is associated with the secretion of dopamine, where approximately 80% of the dopaminergic neurons exist in the basal ganglia. One of the properties of these neurons is that they start firing when a reward is first presented to the primate, but suppress their response with repeated presentations of the same reward stimulus. At this convergent phase, the neurons start responding to stimuli that predicts a reinforcement, i.e. events in the near past that have occurred before the presentation of the reward.

In the early nineties, Barto (Barto, 1995) suggested an actor-critic architecture that was able to facilitate learning based on the properties of the basal ganglia. This architecture gave inspiration to several models that focused on replicating the properties of the dopamine neurons (see Joel et al., 2002 for a review). In the current paper we propose an implementation based on liquid state machines, and demonstrate how the interactions of this region with other neural networks of the brain can be modeled. The proposed implementation follows the actor-critic architecture and is shown in Fig. 4.



**Fig. 4.** The liquid state machine implementation of the actor-critic architecture. Each liquid column is implemented using a liquid state machine with feedforward delayed synapses. The critics are linear neurons, while the readouts are implemented using linear regression. On the top right of the figure (colored with green), we show how the actor-critic architecture is mapped on the model of Fig. 1.

The Critic neurons ( $P1, P2, P3$ ) model the dopamine neurons in the basal ganglia. Their role is to learn to predict the reward that will be delivered to the agent in the near future. The  $A1, A2, A3$  neurons learn based on the signal emitted by the Critic neurons. To model them in the current implementation we use a set of linear neurons. The liquid columns in Fig. 4 encode the input to the basal ganglia circuit. To implement the neurons in each liquid column we use the leaky integrate and fire neuron model:

$$\tau_m \frac{dV_m}{dt} = -(V_m - V_{rest}) + R_m * (I_{syn}(t) + I_{inject} + I_{noise}) \quad (10)$$

where  $V_m$  is the membrane voltage,  $\tau_m = C_m * R_m$  is the membrane time constant,  $R_m$  is the membrane resistance,  $C_m$  is the resistor capacitance,  $I_{inject}$  is a constant current injected to the neuron and  $I_{noise}$  a Gaussian random variable with zero mean and a small variance noise. After the emission of a spike, the membrane potential is reset to its resting value  $V_{rest}$ .  $I_{syn}(t)$  is the incoming current from the presynaptic neurons.

The connections between the neurons in the liquid are implemented using a model of dynamic synapses (Markram et al., 1998). The post-synaptic potential (PSP) of each neuron is transferred to its efferent based on the following equations:

$$PSP_n = L * R_n * u_n \quad (11)$$

$$u_{n+1} = u_n * e^{\left(-\frac{\Delta t}{\tau_{facil}}\right)} + U * \left(1 - u_n * e^{\left(-\frac{\Delta t}{\tau_{facil}}\right)}\right) \quad (12)$$

$$R_{n+1} = R_n(1 - u_{n+1}) * e^{\left(-\frac{\Delta t}{\tau_{rec}}\right)} + 1 - e^{\left(-\frac{\Delta t}{\tau_{rec}}\right)} \quad (13)$$

The maximum output of the synapse is governed by the absolute synaptic efficacy  $L$ . The change of the efficacy is determined using the variables  $u_n$  and  $R_n$ , which are calculated according to eqs. 12 and 13.  $u_n$  defines the utilization of the synaptic efficacy which decays exponentially based on the  $\tau_{facil}$  parameter to its resting value  $U$ .  $R_n$  is the fraction of available synaptic efficacy and defines the strength of the  $PSP_n$  at a given spike. It reduces due to the arrival of new spikes and recovers exponentially according to the  $\tau_{rec}$  parameter.

The actors, i.e. the cortical region that learns based on the predicted rewards of the critics is implemented using a set of linear regression readouts that are trained to output a firing rate proportional to the sum of firing rates of each liquid column. Input from different sources is modeled as a set of rate code neurons that each projects to a separate liquid column using linear synapses with zero delay.

To implement the synapses between the liquid columns and the  $P, A$  neurons, we use the imminence weighting scheme (Barto, 1995). In this setup, the critic must learn to predict the reward of the environment using the weighted sum of past rewards:

$$P_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots + \gamma^t r_1 \quad (14)$$

where the factor  $\gamma$  represents the weight importance of predictions in the past and  $r_t$  is the reward received from the environment at time  $t$ . To teach the critics to output the

prediction of eq. 14 we update their weights using gradient learning, by incorporating the prediction from the previous step:

$$v_t^c = v_{t-1}^c + n[r_t + \gamma P_t - P_{t-1}]x_{t-1}^c \quad (15)$$

where  $v_t^c$  is the weight of the Critic at time  $t$ ,  $n$  is the learning rate and  $x_t^c$  is the activation of the critic at time  $t$ . The parameters  $\gamma$ ,  $P$  and  $r$  are as in eq. 14. The weights of the actor are updated according to prediction signal emitted by the critic:

$$v_t^a = v_{t-1}^a + n[r_t - P_{t-1}]x_{t-1}^a \quad (16)$$

where  $v_t^a$  is the weight of the Actor at time  $t$ ,  $n$  is the learning rate and  $x_{t-1}^a$  is the activation of the actor at time  $t-1$ . In the results section we demonstrate how the output of the Critic neurons approximates the response properties of the dopamine cells discussed above, as well as how the actor neurons learn to control the higher order primitive model.

### Policy learning

Based on the higher order primitives and reward subsystems described above, the problem of reaching can be solved by searching for a policy that will produce the appropriate joint torques to reduce the error:

$$q_e = \hat{q} - q \quad (17)$$

where  $\hat{q}$  is the desired state of the plant and  $q$  is its current state. In practice we do not know the exact value of this error since the agent has only information regarding the end point position of its hand and the trajectory that it must follow in Cartesian coordinates. However because our higher order primitive model is defined in Cartesian space, minimizing this error is equivalent to minimizing the distance of the plant's end point location with the nearest point in the trajectory:

$$d_e = |l - t| \quad (18)$$

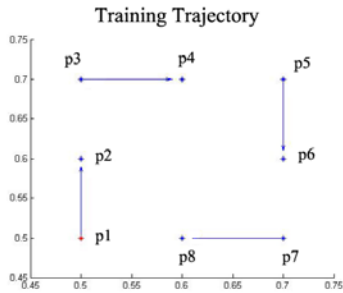
where  $l$  and  $t$  are the Cartesian coordinates of the hand and point in the trajectory, respectively. The transformation from eq. 17 to eq. 18 is inherently encoded in the higher order primitives discussed before.

From the output of the forward model we obtain the end point Cartesian location of the hand, while from the demonstrator we obtain the point in the trajectory that must be reached. These are injected as rate codes into a liquid state machine, where a readout neuron is taught to estimate the subtraction of the two input rates using a feedforward neural network.

The policy is learned based on two elements: (i) decide which higher order primitive force fields will be activated, and (ii) determine each one's weight. The output of the actor neurons described in the previous section implement the activation of the canonical neurons in the F5 premotor cortex which are responsible for gating the higher order primitives. Due to the binary output of the actor neurons, when a certain actor is not firing then its corresponding force field will not be activated. In contrast when an actor is firing, its associated force field is scaled using the output of the subtraction readouts, mentioned above, and added to compose the final movement.



To teach the actors the local control law, we use a square trajectory shown in Fig. 5, which consists of eight consecutive points  $p_1..p_8$ . The agent is taught the trajectory backwards, i.e. starting from the final location ( $p_8$ ) in four blocks. Each block contains the whole repertoire of movements up to that point. Therefore in the first block the actor learns to perform the left motion. Whenever it finishes a trial successfully, the actor is delivered a binary reward, and moves to the next phase which includes the movement it just learned and a new behavior.



**Fig. 5.** The initial trajectory used to train the robot. It consists of 8 points that form 4 perpendicular vectors in four different directions (up, right, down, left).

Reward is delivered only when all movements in a block have been executed successfully. Therefore, the agent learns to activate the correct force field primitives using the prediction signal from the Critic neurons in Fig. 4. The final torque that is applied on each joint is the linear summation of the scaled higher order primitives:

$$\begin{aligned} \tau_p &= [x_{e,1} * (J^{-1})^T * \varphi_{up}]_{act1} \\ &+ [x_{e,2} * (J^{-1})^T * \varphi_{down}]_{act2} \\ &+ [x_{e,3} * (J^{-1})^T * \varphi_{right}]_{act3} \\ &+ [x_{e,4} * (J^{-1})^T * \varphi_{left}]_{act4} \quad (19) \end{aligned}$$

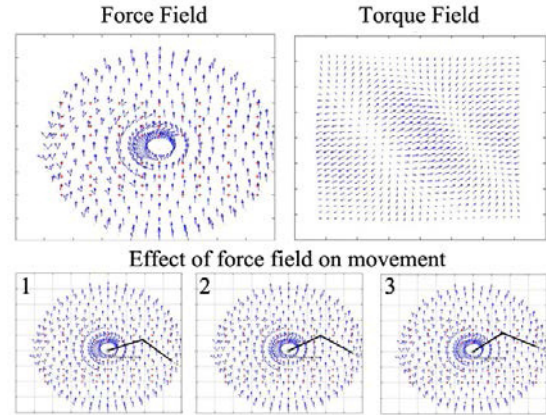
where  $x_{e,i}$  is the output from the neural network distance readout, while  $[ ]_{act}$  is an operator that includes each force field in eq. 19 only if the corresponding actor from the basal ganglia module is active.  $\varphi$  is obtained from eq. 6 for each higher order force field, and  $J$  from eq. 7.

## Results

In the current section we present the results of the proposed model. We focus on the training of each pathway, as well as the model's ability to follow various different trajectories.

The first result we consider is the convergence of the least squares solution for the system of linear equations in eq. 9. Figure 6 presents the solution for the “up” higher order primitive, where it is evident that the least squares algorithm has converged to a good result. The three subplots at the bottom illustrate how the hand moves towards the “up” direction when this force field is active. Similar solutions were obtained for the other three primitives, where the least squares solution converged to 7 (left), 2 (right) and 5 (down)

errors (the error represents the average deviation of the vectors in a field from the correct direction of the force).

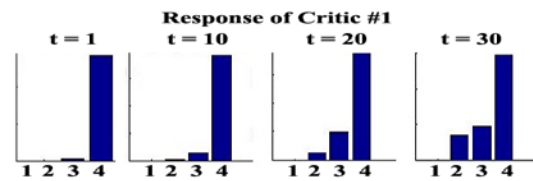


**Fig. 6.** The force field (upper left subplot) and torque field (upper right subplot) as converged by the least squares solution for the “up” primitive. The three subplots at the bottom show how the hand moves when the primitive is active.

The policy for reaching was learned during the initial imitation phase described previously. During this phase the robot performed the training trajectory, and was delivered a binary reinforcement signal upon successful completion of a whole trial.

Since the reward signal was only delivered at the end of the trial, the agent relied on the prediction of the reward signal elicited by the critic. In the following we look more thoroughly on the response properties of the simulated dopaminergic critic neurons and how the actors learned to activate each force field accordingly based on this signal.

Figure 7 illustrates how the critic neurons of the model learned to predict the forthcoming of a reward during training. In the first subplot (first successful trial) when reward is delivered at  $t=4$ , the prediction of the 1<sup>st</sup> critic is high, to indicate the presence of the reward at that time step. After the first 10 successful trials (Fig. 7, subplot 2), events that precede the presentation of the reward ( $t=3$ ) start eliciting some small prediction signal. This effect is more evident in the third and fourth subplots where the prediction signal is even higher at  $t=3$  and starts responding at  $t=2$  as well.

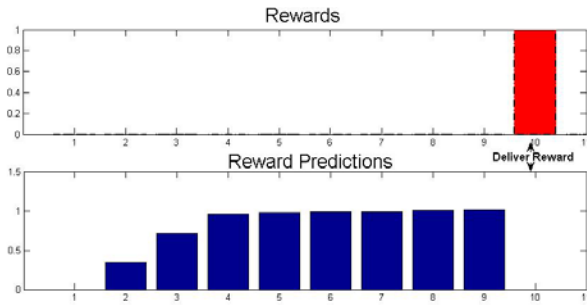


**Fig. 7.** The prediction signal emitted by the critic component of the model during the initial stages of the training (subplot 1), after 10 trials (subplot 2), after 20 trials (subplot 3) and after 30 trials (subplot 4).

The effects of this association are more evident in Fig. 8, where it is shown that, after training, even though rewards are not available in the environment, the neurons start firing because they predict the presence of a reward in the subsequent steps. Using the output of this prediction signal,

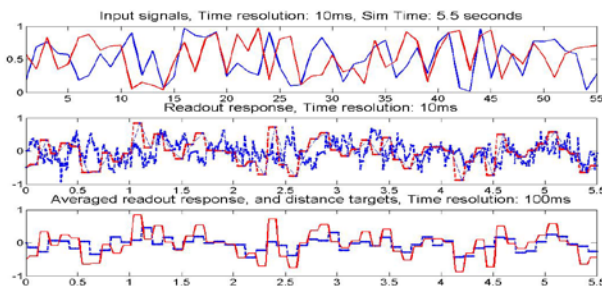


the actor, i.e. in the case of the model the F5 premotor neurons that activate the force fields in the MI, forms its weights in order to perform the required reaching actions.



**Fig. 8.** The actual reward signal given to the robot at the end of a successful trial (upper subplot), and the reward predicted by the critic component after training (bottom subplot). The x-axis represents the 100ms time blocks of the simulation while the y-axis the values of the reward and prediction signals respectively.

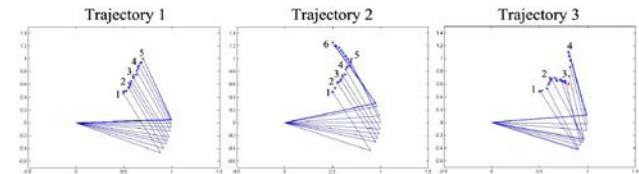
The second part of the policy is for the model to learn to derive the distance of the end effector location from the current point in the trajectory. This is accomplished by projecting the output from the forward model and perception pathways in an LSM and using a readout neuron to calculate their subtraction. Having run several different simulations we found that to shape the liquid dynamics and learn this transformation the dynamic synapses must have delays of approximately 10ms. Since our model resolution was set to 100ms, we averaged the output of the readout neuron over the 10 steps of the simulation. In Fig. 9, we illustrate two sample signals as input to the liquid (top subplot), the output of the readout neuron in the 10ms resolution (middle subplot) and the averaged over the 100ms of simulation time output of the readout neuron (bottom subplot).



**Fig. 9.** The output of the distance LSM after training. The top plot illustrates two sample input signals of 5.5 seconds duration. The bottom two plots show the output of the neural network readout used to learn the subtraction function from the liquid (middle plot), and how this output is averaged using a 100ms window (bottom plot).

The whole simulation trial lasted 5.5 seconds. As the results show the liquid was able to extract the distance information with a good accuracy. Due to the local control laws used to implement the reaching policy, any small errors in the computation of distance are actually compensated in later steps.

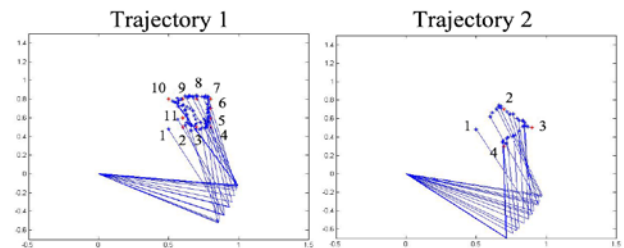
Having established that the individual pathways/components of the proposed model operate successfully, we now turn our attention to the performance of the model in various reaching tasks. We note here that the model wasn't trained to perform any of the given reaching tasks, apart from the initial training/imitation period at the beginning of the experiments, shown in Fig. 5. After this stage the model was only given a set of points in a trajectory and followed them with very good performance. The first three trajectories we tested were variations of a straight line motion.



**Fig. 10.** Three trajectories shown to the robot (red points) and the trajectory produced by the robot (blue points). Numbers mark the sequence with which the points were presented.

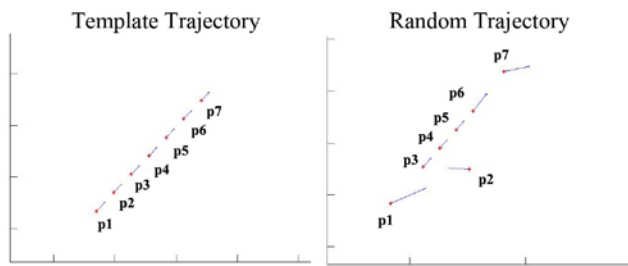
As Fig. 10 shows the agent was able to follow all three trajectories quite precisely. The average normalized deviation of the agent's position from the points of the trajectory was 0.03 which shows that the resulting performance was satisfactory.

In order to evaluate further the performance of the model we used two more complex trajectories. The first required the robot to reach towards various random locations spread in the robot's workspace (Fig. 11, Trajectory 1) while the second complex trajectory required the robot to perform a circular motion in a cone shaped trajectory (Fig. 11, Trajectory 2). Figure 11 illustrates how the aforementioned trajectories were followed by the robot.



**Fig. 11.** Two complex trajectories shown to the robot (red points) and the trajectories produced by the robot (blue points). Numbers mark the sequence with which the points were presented.

To evaluate the performance of the model on any given path we created 100 random trajectories and tested whether the agent was able to follow them. Each of these random movements was generated by first creating a straight line trajectory (Fig. 12, left plot) and then randomizing the location of 2, 3 or 4 of its points; an example is illustrated in Fig. 12, right plot. The error was calculated by summing the overall deviation of the agent's movement from the points in the trajectory for all the entries in the dataset. The results indicate that the agent was able to follow all trajectories with an average error of 2%. This suggests that the selected model can confront with high accuracy any reaching task.



**Fig. 12.** The template used to generate the random test set of 100 trajectories (left plot) and a random trajectory generated from this template (right plot).

## Conclusion

One of the important aspects of human skills is the ability to generalize knowledge to different domains and tasks. Using modularity and principles from neuroscience, in the current paper we investigated how adaptive learning skills can be acquired in a simulated agent that performs reaching tasks. One of the extensions that we plan for the presented model is to investigate how the primitive model can be designed to be adaptive, i.e. allow the agent to match the control of primitives to the properties of its body. In addition we will extend the current 2D plant model to its 3D equivalent by adjusting the equations of the Jacobian and primitive model. Moreover, we plan to investigate the role of the cerebellum in reaching movements, and its involvement in providing corrective feedback in respect to the global error of movement. Finally one of the important additions that we plan to investigate is how the agent developed in the current paper can be used during observational learning, i.e. improve its performance in reaching tasks without using its body. This extended model will be used to evaluate certain hypotheses regarding whether learning can be implemented in primates during observation.

## References

- Atkeson (1989). Learning arm movement kinematics and dynamics, *Annual Review of Neuroscience*, 12(1):157-183
- Ballard D.H. (1986). Cortical connections and parallel processing: Structure and function, *Behavioral and Brain Sciences*, 9(1):67-90
- Barto, A.G. (1995). Adaptive critics and the basal ganglia, *Models of information processing in the basal ganglia*, MIT Press, Cambridge.
- Bizzi E., Mussa-Ivaldi, F.A. and Giszter, S. (1991). Computations underlying the execution of movement. A novel biological perspective, *Science*, 253(5017):287
- Degallier, S. and Ijspeert, A. (2010). Modeling discrete and rhythmic movements through motor primitives: a review, *Bio. Cyb.*, pp.1-20
- Doya, K. (1999). What are the computations of the cerebellum, the basal ganglia and the cerebral cortex, *Neural Networks*, 12(7):961-974.
- Dum, R.P. and Strick, P.L. (1991). The origin of corticospinal projections from the premotor areas in the frontal lobe, *Journal of Neuroscience*, 11(3):667
- Giszter, S.F. and Mussa-Ivaldi, F.A. and Bizzi, E. (1993). Convergent force fields organized in the frog's spinal cord, *The Journal of Neuroscience*, 13(2):467
- Hourdakis E. and Trahanias, P. (2011). Improving the performance of liquid state machines based on the separation property, submitted to the *Engineering Applications of Neural Networks*, EANN11, Corfu.
- Hourdakis E. and Trahanias, P. (2009). A framework for automating the construction of computational models, *CEC2009*, Norway
- Jeannerod, M., Arbib, M.A., Rizzolatti, G. and Sakata, H. (1995). Grasping objects, the cortical mechanisms of visuomotor transformation, *Trends in Neurosciences*, 18(7):314:320.
- Joel, D., Niv, Y. and Ruppel, E. (2002). Actor-critic models of the basal ganglia: a new anatomical and computational perspectives, *Neural networks*, 16(4):535-547.
- Kaas, J.H., R.J. Nelson, M. Sur, C.S. Lin, and M. M. Merzenich. (1979). "Multiple representations of the body within the primary somatosensory cortex of primates". *Science* 204(4392 ): 521-523.
- Maass, W., Natschlager, T. and Markram, H. (2002). Real Time computation without stable states: A new framework for neural computation based on perturbations, *Neural Computation*, 14(11): 2531-2560
- Markram H., Wang, Y. and Tsodyks, M. (1998). Differential signaling via the same axon of neocortical pyramidal neurons, *Proceedings of the National Academy of Sciences*, 95(9):5323
- Mountcastle, V.B. (1978). An organizing principle for cerebral function: The unit module, *The mindful brain*, MIT Press, Cambridge
- Paul R. P. (1981). *Robot Manipulators: Mathematics, Programming, and Control*. Cambridge, Massachusetts: MIT Press
- Piaget (1962). *Dreams and imitation in childhood*, Norton, New York
- Plaut, D.C. and Hinton, G.E. (1987) Learning sets of filters using backpropagation, *Computer Speech and Language*, 2(1):35-61
- Sakai, K., Hikosaka, O., Miyauchi, S., Takino, R., Sasaki, Y. and Putz, B. (1998). Transition of brain activation from frontal to parietal areas in visuomotor sequence learning, *Journal of Neuroscience*, 18(5):1827.
- Sergio, L.E. and Kalaska, J.F. (2003). Systematic changes in motor cortex cell activity with arm posture during directional isometric force generation, *Journal of Neurophysiology*, 89(1):212.
- Soechting, J.F. and Flanders, M. (1992). Moving in three dimensional space: frames of reference, vectors and coordinate systems, *Annual Review of Neuroscience*, 15(1):167-191
- Thach, W.T., Mink, J.W., Goodkin, H.P. and Keating, J.G. (2000). Combining versus gating motor programs: Differential Roles for the cerebellum and basal ganglia?, *Cog. Neuros.: A reader*, pp.366-375
- Todorov, E. (2000). Direct control of muscle activation in voluntary arm movements: A model, *Nature Neuroscience*, 3(4):391-398.
- Wolpert, D.M. (1997). Computational approaches to motor control, *Trends in cognitive sciences*, 1(6):209-216.

# Energy as a driver of diversity in open-ended evolution

Tim Hoverd<sup>1</sup> and Susan Stepney<sup>1</sup>

<sup>1</sup>YCCSA, University of York, YO10 5GE, UK  
tim.hoverd@cs.york.ac.uk

## Abstract

We investigate the consequences of introducing an energy model into open ended evolutionary simulations. We propose a metamodel for simulations that incorporate an energy model and apply that model by extending Turk's *Sticky Feet* model. We show that introducing an energy model produces simulations with measurably increased diversity of the simulated population.

## Introduction

We are interested in open ended evolution and in particular evolution within systems that are open to a simulated energy flux, open to changes in the simulated environment, and open to the representation of evolutionary mechanisms. In this paper we focus on energy flux, which allows us to represent many aspects of real world systems, such as the availability of food supplies, and different means of making a living within an environment, be they predatory or sessile.

In order to investigate these issues we have chosen to extend Turk's *Sticky Feet* [10] model. This gives a simple mechanism for implementing mobility and experimenting with open-ended evolution. A *Sticky Feet* simulation is a collection of simulated creatures moving in a 2D domain. Each such creature is a graph of *springs* connecting together *feet*. Motion is achieved as a consequence of simple harmonic oscillation of the springs, which pushes the feet around within the simulation space. The coefficient of friction experienced by the feet is modulated—at times slippery, at times sticky—which results overall in motion through the space.

Each creature has a heart and a mouth, each of which is a distinguished type of foot. The heart represents the creature's 'essence'. The mouth—when it happens upon another creature's heart—allows the former creature to eat the latter, removing it from the simulation. The likelihood of a creature happening upon another is facilitated to some extent by the springs being equipped with sensors, which may modulate the oscillation of the spring when in the presence of another creature's heart. This allows a creature to turn towards another, with the chance that it might then be able to

consume the target. When a creature is consumed the eater produces a single offspring, which may be a mutation of the parent. Mutations that include additional feet, springs and sensors allow the creatures to evolve in a manner that eventually produces offspring that are better adapted to hunting for and eating other creatures.

A *Sticky Feet* world is one in which creatures evolve to improve their performance at consuming other creatures, and therefore being able to pass on their genome. As such, it provides some aspect of a model of open ended evolution. We use this term here in the sense of an evolutionary system where components continue to evolve new forms continuously, rather than halting when some 'optimal' or stable position is reached [9].

*Sticky Feet* [10] works in this manner, as there is no overall fitness function and all creature behaviour is expressed in a single large environment rather than relying on artificial two-creature tournaments. As such it is representative of many aspects of real-world evolution.

There is, though, no mechanism for sticky feet creatures to pass on their genomes other than by consuming other creatures. That is, the simulation is closed to the development of non-predatory behaviour. This is useful from the point of view of maintaining a constant sized simulation, but is not representative of real world evolution where population sizes can change dramatically.

Natural evolution—that which operates in the world around us—is different in essence from the sticky feet model in that success does not entirely derive from hunting and reproduction. Creatures in natural environments must be able to extract some sort of living from that environment, supported either by consuming other creatures, or by turning some flux in the world, for example sunlight or the chemical nutrients consumed by extremophiles, into food.

This argument is essentially that famously made by Malthus in 1798 [7], which led Darwin towards the principle of natural selection [2]. Although Malthus discussed the availability of food we generalise this to the availability of *energy*. This is a limited resource although the environment is continually bathed in an energy flux. This flux may

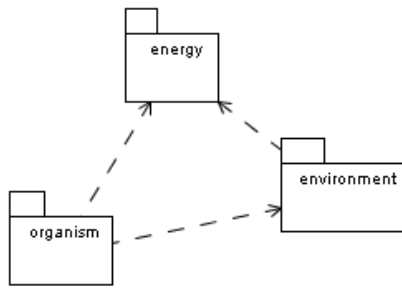


Figure 1: Domain metamodel

be used, and stored, by components of the environment, but if it is ignored it disappears and is no longer of use.

Natural systems are open: they are in receipt of some sort of resource flux such as that we model as energy. In this paper, we provide a meta-model for open simulations with energy flux, consumption, and storage; we describe an extended sticky feet simulation incorporating an instantiated energy model; we show that diversity is maximised when the flux is neither too low, nor too high.

## Energy metamodel

In our work we use the CoSMoS approach [1]. We model the aspects of the domain that we wish to simulate as the *domain model*. We describe the actual simulation using the *platform model*, which executes on the *simulation platform*, producing results that can be analysed with respect to the *results model*.

In this paper we describe a *class* of models, ones that permit a particular sort of open ended evolution of sticky feet like creatures in a world, a domain, which is bathed in an energy flux. That is, we must define a *domain metamodel* to which our domain models must conform.

The domain meta model describes all possible domain models that we wish to explore, without limiting the particular domain. An abstract view of our meta-model is shown as figure 1<sup>1</sup> and shows the inter-dependencies of the three top level packages in our model: *Organism*, *Energy* and *Environment*.

## Energy

Energy is modelled, as in figure 2, as a scalar quantity in arbitrary units. We also describe the *entropy* of some energy which might be thought of as the temperature of the energy which allows us to describe essential aspects of the energy economy. For example, in the natural world a continuous low flux of low entropy energy is available in the form of sunlight. Plants sequester this energy in a form that allows other organisms, such as animals to consume them and acquire the stored energy. Those animals subsequently excrete waste products which still represent energy, albeit in

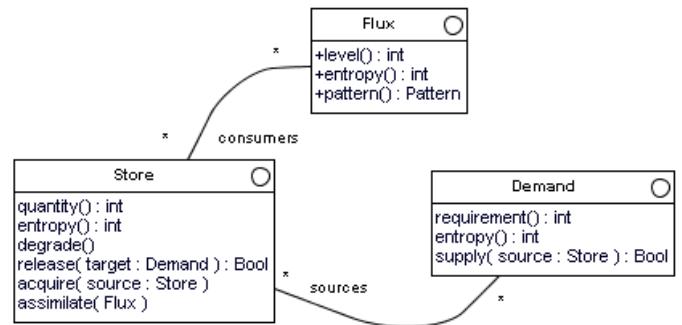


Figure 2: Energy metamodel

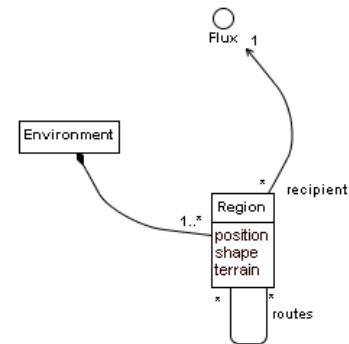


Figure 3: Environment metamodel

a higher entropy form, but which may still be metabolized by organisms such as dung beetles.

Although some authors use a simple "battery" model or conservation of energy (for example [3]), here we propose an energy model integrated with reproduction and behaviour.

**Flux.** The most basic part of the energy model, representing a flow of energy from outside the modelled system. This flux represents energy with a defined entropy and with a particular temporal pattern; for example at a high level during daytime but a much lower level during nighttime.

**Store.** One action of all members of a simulated world is to store energy. An organism might maintain its existence by consuming other stores, in the manner of herbivores eating plants, or by assimilating the flux itself as the plant itself does.

**Demand.** Many components of a simulated world make energy demands. Such components could be the physical structure of an organism, which requires energy to build and maintain, or an activity that an organism undertakes, such as hunting for other organisms to consume.

## Environment

The environment metamodel is elaborated in figure 3.

<sup>1</sup> All the models here are expressed using the UML.

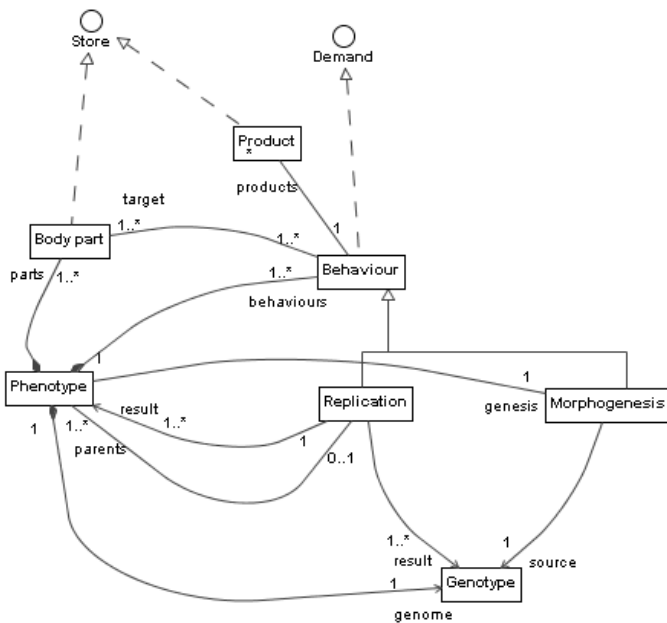


Figure 4: Organism Metamodel.

Environments are represented as a collection of Regions, each of which is the recipient of a particular Flux. Regions are connected together by routes each of which allows organisms to move from one region to another, albeit at a certain energy cost.

## Organism

The organism metamodel is elaborated in figure 4<sup>2</sup>. It has two interdependent components: the phenotype and the genotype.

**Genotype.** The Genotype metamodel requires that an organism model expresses a genotype, which can be used as the source information for a morphogenesis process that grows its associated Phenotype. The genome of a phenotype is the result of a replication process that also creates a new Phenotype.

**Phenotype.** The Phenotype metamodel expresses that an organism's phenotype, its structure, consists of a number of *body parts* and a number of *behaviours*.

Body parts store energy: they realise the Store component of the energy model. The body parts are also the *target* of the organism's behaviours. For example, a bird's wings might be the target of its 'flying' behaviour. Each behaviour affects at least one part of an organism's body, and all such parts must be such a target of at least one behaviour.

Behaviours consume energy: they realise the Demand component of the energy model. We require that all energy

consumption is expressed as a behaviour. So, for example, a purely sessile organism must still include a behaviour that it continually expresses, which demands the energy needed to maintain its metabolism. The energy for a behaviour is supplied by the body parts that are the target of the behaviour.

Some of an organism's behaviours produce waste products, included as the *Product* component. Such waste products are in themselves further energy stores, although they are not part of the organism's phenotype. The entropy of such waste products would usually be higher than that of the original energy source, but that does not preclude some organisms being able to scrape out an existence using such low grade sources of energy. A further waste product is the phenotype of a dead creature. Again this represents a low-grade source of energy, providing carrion-eating as a possible way of making a living in a world that conforms to our model.

All organisms possess the *Morphogenesis* behaviour; the genome contains the information needed for this behaviour. The specific genome of an organism is the result of another behaviour, *Replication*, which creates the genome of an offspring organism, potentially generating a mutated genome.

## Discussion

Our metamodel expresses the essential requirements for evolution in an energetic context. A range of different implementations of this model are feasible. That is, a number of models could be produced, each of which conformed to this metamodel in the sense that the model's components were instances or realisations of components in the metamodel. Each such model would describe the domain model for a particular set of simulations in a particular domain.

Note that some ALife simulations incorporate a very basic notion of a constrained resource. Tierra [8] uses CPU time-slices as an analogue of energy, with the size of the time slice being a tunable function of the entity's size. However, there is no analogue of an energy store that would enable entities to 'time-shift' their use of the resource, or hand on a surplus to their progeny; Tierra is a 'use it or lose it' model. (Ray [8] mentions a possible extension allowing capture of CPU slices.) Stringmol [5] is an AChem with an explicit, but very simple, energy model: a fixed number of energy units are added to the container at each timestep, and molecules need to use an amount to execute each instruction. However, the energy is a global resource (energy is not stored in individual entities, but in the system and accessible to all). Our rich energy metamodel provides a number of features that organisms should be able to exploit to enable a range of different ways of making a living.

## Energetic sticky feet

We have developed one simulation model (figure 5) that conforms to our energy metamodel. This is an 'energetic' variant of Turk's Sticky Feet [10]. That is, our energetic sticky feet model discusses the same sort of concepts that Turk

<sup>2</sup>The arrowheads in this diagram refer to the UML property of *navigability* not to a notion of one object "producing" another.

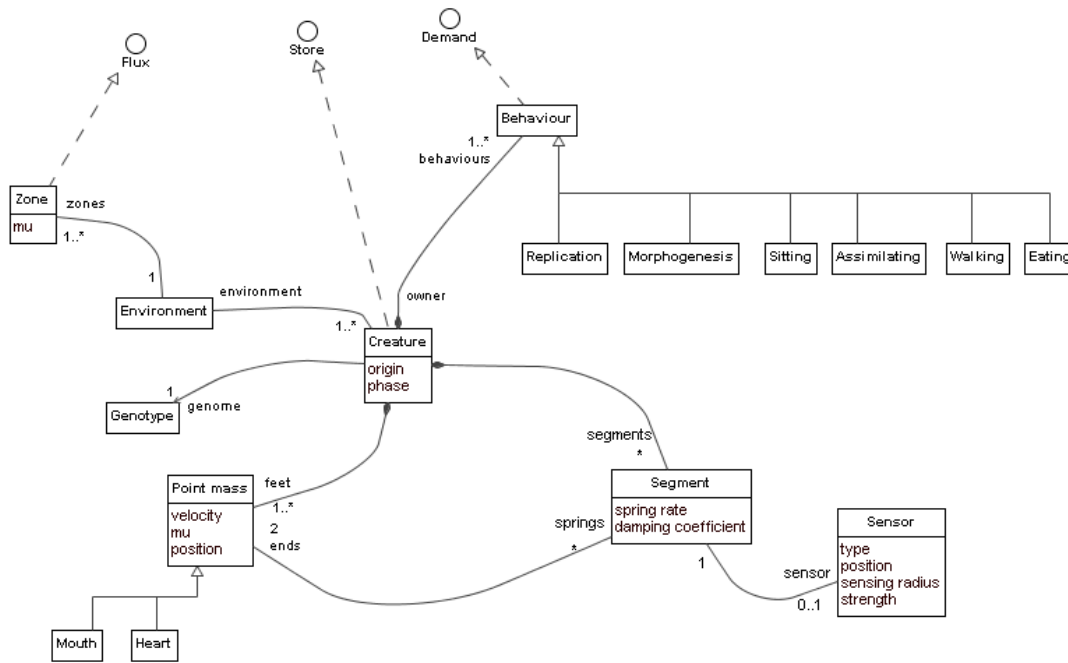


Figure 5: Energetic sticky feet overview

uses, albeit in the context of energy, environment and organism as prescribed by our metamodel.

Our experimental hypothesis is that the presence of the energy model will influence the evolution of the simulated creatures in such a manner that a more diverse world will result. We test this hypothesis by running the energetic sticky feet simulation for a range of flux levels, and compare the diversity of the evolved creatures with an unconstrained variant that ignores the need for energy.

## Body parts

The creatures in our model follow the metamodel: each creature has a number of parts and a number of behaviours. The specific body parts are *feet* and *segments*. Following Turk [10]: a foot is a point mass with a particular, and modulatable, coefficient of friction; a segment is a spring that follows the equations from [10], to achieve motion due to simple harmonic oscillation of the springs as the point masses' coefficients of friction are varied.

The feet themselves appear in three varieties. The basic ones are augmented with special variants, representing a heart, and a mouth. The heart represents the 'essence' of a creature. When one creature's mouth gets close to the heart of another creature then the former may 'eat' the latter (assuming that the former creature is expressing the 'eating' behaviour).

Each segment may optionally have an attached sensor, which senses the position of other creatures. A sensor may sense either the heart or the mouth of another creature and, when it does, may perturb the oscillation of its attached seg-

ment. In this manner the sensors allow a creature to turn towards prey, or away from a predator.

## Behaviours

The overall behaviour of each creature is represented by attaching a collection of individual behaviours to the creature. Each of these acts in a manner reminiscent of the *Command* pattern [4], and applies itself if it determines that the time is appropriate. Every behaviour demands energy, which must be provided by the owner of the behaviour. If the owner cannot supply the energy then the creature dies: it has exhausted its energy supplies.

Our energetic sticky feet implementation does not implement the waste product component of the metamodel. Consequently, when a creature dies, it just disappears from the simulation, taking with it any residual energy.

The behaviours available to a sticky creature are:

**Sitting:** the 'null' behaviour that all creatures must express. This behaviour forces a creature to continually consume energy. The amount of energy consumed is a function of the complexity of the creature's phenotype; a larger, more complex, creature requires more energy just to sit in one place compared to a small, simple, creature.

**Walking:** the behaviour that expresses the mode of walking explored by Turk [10], by oscillation of the creature's segments. The size of the energy requirement is proportional to the friction against which work is done by the springs.



**Eating:** the behaviour that allows a creature to look to see if any other creature's heart is in the vicinity of one of its mouths. If so, the former creature may 'eat' the latter. This adds to the eating creature's energy stores all of the energy of the eaten creature. The eaten creature is removed from the simulation.

**Reproducing:** the behaviour that allows a creature to create offspring, with a genome that is a mutation of the single parent's genome. At each simulation step there is a probability, encoded in the genome, that a creature may express this behaviour. We allocate energy costs to all the components of the phenotype, and check that the parent has sufficient energy to construct the child organism. If so, and the child organism is deemed to be viable, then it is created and the energy store of the parent is shared equally between the parent and the child.

**Morphogenesis:** the behaviour that is followed to construct the phenotype of a new organism from the genome generated by, optionally, mutating the genome of the organism's parent. This differs from the Reproducing behaviour in that it is responsible for building the phenotype of the organism from its genome whereas the reproducing behaviour creates the new organism's genome.

**Assimilating:** the behaviour that allows an organism to gather energy directly from the flux in the current environment. The amount of energy available is determined by the flux applied to the region of the environment that the creature is inhabiting, and by the physical size of the creature. A larger creature, in the same manner as a large tree, can extract more energy from the flux, but needs correspondingly more energy to construct and maintain the larger phenotype.

## Mutation and morphogenesis

In order to get some sort of evolution of the sticky feet creatures our implementation allows for mutation of the genome whenever the reproducing behaviour is expressed. Mutation is implemented by structuring a genotype as a sequence of genes, each of which codes for a particular part of the creature and its behaviour. Unlike Turk [10] we do not express a 'species' in any way in our model. Rather, each organism just has its own genome; even though it is likely that many other creatures have the exact same genome we do not use this in any part of our simulation. Following Turk's lead we implement two general forms of mutation, both of which are used in any individual mutation step. The first of these is the modification of the various parameters that apply to each component. For example this allows the position of the creature's feet, the stiffness of the springs in the segments, and the probability that a creature will attempt to express the reproducing behaviour at any particular point in time to be varied. The second form represents structural modifications of the phenotype. Specifically, these modifications may be

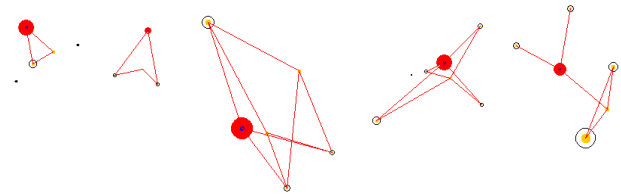


Figure 6: Some example evolved creatures; the filled circle is the heart, the open circles are mouths. From left to right these are: a) the initial 'seed' creature; b) the 'manta ray', only a few mutations away from the seed; it has two mouths; c) the 'killer', large and fast; d) the 'multimouth', with lots of mouths that stab outwards; e) the 'spiky', with lots of mouths but little area.

performed: adding feet or segments, removing feet or segments, adding a sensor to a segment and modifying a segment so as to connect to a different foot.

A possible result of one or more of these mutations is that the eventual creature does not form a viable phenotype. For example, it is possible to generate a genome that implies a phenotype where the feet and segments are not connected as a single structure, or where a creature does not have a heart. We choose to declare these mutations non-viable, and terminate the particular cycle of reproduction when they occur.

Even if a mutation represents a viable creature, it is possible that the resulting creature cannot be incorporated into the current simulation world. Specifically, in a similar manner to Turk, we do not allow phenotypes that initially overlap existing creatures. That is, our simulations are expressly two dimensional at the moment.

Viable creatures are created at a point in the simulation space that is local to their parent.

## Implementation

Our energetic sticky feet implementation follows closely the model shown in figure 5. The implementation is written in pure Java and uses our environment-orientation approach [6] to represent the interaction of many creatures in a multi-threaded implementation. The environment is a two dimensional world with cyclic boundary conditions.

In all our experiments we initialise a simulation run with a fixed number of simple 'seed' creatures with a pre-defined genome and a random (according to a Gaussian probability distribution function) amount of energy. A typical collection of evolved creatures is shown in figure 6.

There are a large number of parameters to our sticky feet simulations. For example, there are parameters describing the construction energy required for each part of an organism, for the rate of mutation and for the level of energy flux in different regions of the environment.

Initial experiments with our implementation show that careful setting of these parameters is necessary in order to



allow the creatures to survive. That is, it is very easy to set the parameters so that there is insufficient energy in the environment for a population of creatures to survive; even though they can mutate to take advantage of their environment they run out of time in which to do so. This is in some ways perhaps a consequence of our approach of seeding the environment with a collection of fully formed creatures with significant energy demands.

## Experimentation

In order to compare our simulations with something more representative of Turk's implementation [10] we need a way of 'turning off' the energy model. That is, we need to be able to run simulations in a manner that is not constrained by the availability of energy. In Turk's implementation the simulation has a fixed size population as a consequence of each creature reproducing once only when it consumes another. Hence, the simulation world does not get overrun with a vast number of creatures.

In a similar manner, our simulation includes an 'unconstrained energy' option where the creatures function exactly as they do in the energetic world except that the *demand* of all behaviours is set to zero, so no energy is ever consumed, and the *reproducing* behaviour is only available, and indeed is forced, in the situation where the *eating* behaviour has been invoked. This has the effect of creating a fixed-population simulation (except that on occasion a new creature cannot be 'fitted in' to the existing simulation, in which case reproduction is delayed until space is available) of a form similar to Turk's.

The differences between the implementation of the 'energetic' and 'unconstrained' variants of our simulation are minor. Hence, we can be sure that measured differences in the results of the simulations are a consequence of the inclusion, or exclusion, of the energy model.

In order to track the develop of creatures as they evolve we use a notion of *mutation distance* in our experiments. As discussed we have no specific notion of 'species' in our implementation. Rather, each creature has its own genome, which has a mutation distance. The initial population of creatures all have a copy of the same genome, which has *mutation distance* = 0. Whenever a creature reproduces it may also mutate the genome which is passed on to the child creature. The likelihood of allowing such a mutation is one of the simulation's parameters. After this mutation, following the process described earlier, the implementation compares the resulting genome with the initial genome. If they are different (they might not be because of the random nature of choosing whether to adopt specific mutations) and the genome represents a viable creature, then that new genome mutation distance is incremented.

In this manner every creature has a mutation distance, and we use this as part of our experimental results. There is not a simple relationship with *time*; it is possible, although un-

likely, for example, for a creature with mutation distance 150 to co-exist in a simulation with another of mutation distance 0. The latter creature could have survived from the outset—our creatures do not die of old age—or it could be the end result of a series of reproductions that involved no mutations.

Each creature in our simulations has an *area* that determines the amount of energy it receives from the environment's energy flux. We calculate its area by regarding the creature as an irregular polygon, ignoring feet that have only one attached segment, and by calculating the area of that polygon. So, a creature that was two feet connected by a single segment (a frequently occurring shape) would have  $area = 0$  and would not receive any energy from the environmental flux.

## Hypotheses

The direction of our experimentation is towards investigating two hypotheses. First, we hypothesise that creatures evolving in the context of an energy model should do so in a manner that is measurably different from that which applies in a 'unconstrained energy' world. Second, we hypothesise that the presence of the energy model creates a wider range of ways of the sticky feet creatures 'making a living'. For example, a creature could survive by eating other other creatures, or it could survive by growing large enough to acquire sufficient energy from the regional flux. Such a mode of life could be further enhanced by abandoning movement as that could be seen as wasting precious energy. Hence, we hypothesise that when evolving in the presence of an energy model the sticky feet creatures will appear in a wider range of sizes during their evolution than happens in an 'unconstrained energy' world.

Similar hypotheses could be expressed about other physical aspects of the creatures. Here we explore just the size.

## Results

Our simulations generate a large quantity of data and here we show just a single summary of one aspect of it. Figure 7 shows a plot of the inter-quartile range of the sizes (areas) of the population of creatures as it changes with the genome mutation distance. This figure includes data for three different configurations: the unconstrained 'control' situation, one with an energy flux of 80 (arbitrary) energy units, and one with an energy flux of 100 units. Data for this plot are taken from a total of over 40 separate simulation runs and summarise the simulated lives of over 350,000 energetic sticky feet creatures.

We have chosen these energy levels based on experience running our simulations. Below an energy flux of 80 units it is invariably the case that the population of creatures dies out. For example, in all our experimental data no creature has existed in a simulation with a flux of 70 with a higher

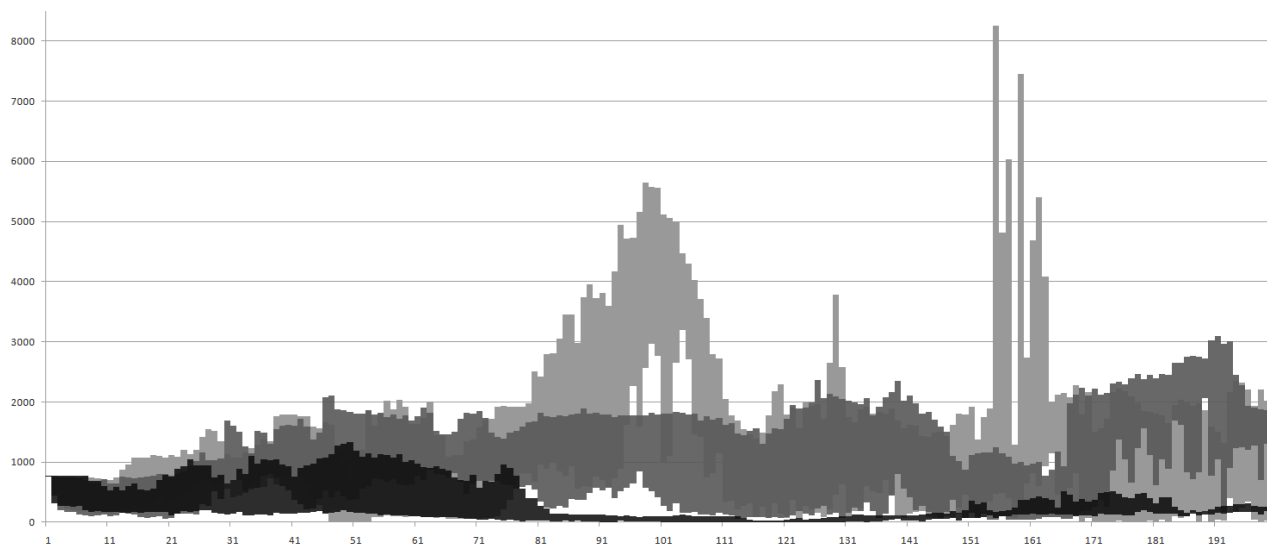


Figure 7: Summary of results of execution of energetic sticky feet simulation with mutation distance on the X axis. The foreground, darkest, distribution shows the inter-quartile range of the areas of sticky feet creatures across 200 mutation distances of evolution using the implementation that did not use an energy model. The mid-grey plot is the same information using the energy model at a flux level of 100 units. The pale grey plot shows the results using the energy model at a flux of 80 units.

mutation distance than 94. At a flux of 50, we see nothing beyond mutation distance 73.

Even without comparing the data with the unconstrained situation we see a clear effect of the flux on the simulated lives of the sticky feet creatures. Furthermore, inspection of figure 7 shows significant differences between the ‘with energy’ and ‘unconstrained energy’ variants of our simulation. For example, at mutation distance 200 (the largest we show on the figure) creatures in the *energy* = 80 world have a range of sizes from 280 area units at the lower quartile to 2170 units at the upper quartile. In the ‘unconstrained energy’ world the equivalent sizes are from 140 to 238 units.

Experience with our experiments, and observation of the results shown here, leads us to a further hypothesis. This is driven by the observation, seen in figure 7, that at *energy* = 100 there is less population diversity than at *energy* = 80. As we know that at lower energy levels the populations of sticky feet creatures usually dies out we hypothesise that there is a critical energy flux density, in a set of simulations with otherwise consistent parameters, that generates creature populations of the widest diversity. At low energy levels there is insufficient energy for populations to survive and hence they die out before generating significant diversity; at higher energy levels it becomes easier and easier to make a living, all the way up to the unconstrained world.

We choose a single statistic to represents diversity of simulations with a particular energy flux, and look to see if it varies in the hypothesised manner. The statistic we use is the range of sizes of creatures throughout all lifetimes at a particular energy level. Figure 8 is a box and whisker plot

of the interquartile range (IQR) of sizes of creatures over all mutation distances. In figure 8, the median represents the median IQR of sizes over mutation distance at a particular energy flux (the median size of the bars in figure 7): the larger the median, the larger the range of sizes, hence the greater the diversity. In figure 8, the IQR represents the variation in the IQR of sizes over mutation distance at a particular energy flux (the range of sizes of the bars in figure 7): the larger the IQR in figure 8, the larger the range of range of sizes, hence the greater the range of diversity. Observation of figure 8 does indeed show the hypothesised characteristic of a critical energy flux with maximum diversity.

## Discussion

The hypotheses that we have discussed are supported by the experimental results we have included. Specifically, the results we see when running the ‘energetic’ simulations show a more diverse range of creatures being produced than in similar ‘unconstrained energy’ situations. Furthermore, there is a ‘critical’ energy level that supports the widest diversity. At lower energies we see less diverse populations that soon die out; at higher energies—which includes the unconstrained case—we see less diverse populations that nonetheless persist. The critical energy level is the point between a low energy world where eating other creatures is a necessity of life, but nevertheless there is not enough influx of energy to survive, and a high energy world where there is little evolutionary pressure, and sessile behaviour is common.

Visual inspection of our simulations make it painfully

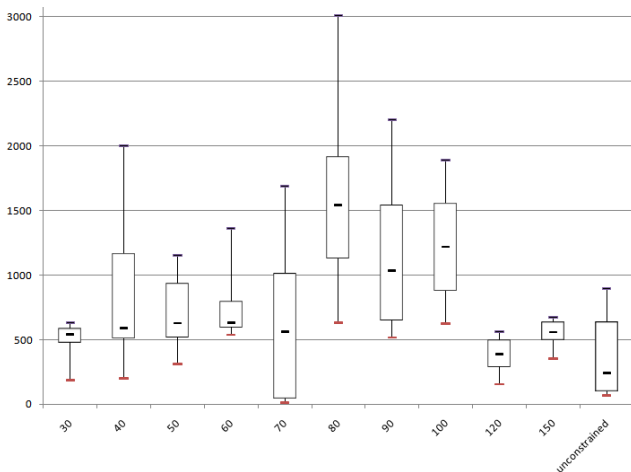


Figure 8: Box plot of inter-quartile ranges of creature sizes at various energy levels. Each plot shows the 9th percentile, the first quartile, the median, the third quartile and the 91st percentile for the distribution of the inter-quartile ranges at the given energy flux. The rightmost box is for the unconstrained energy version of the simulation.

clear that although we generate creatures with a wide range of sizes and structure they are still recognisably the same sort of thing: variations on a theme of feet and springs (figure 6). The end result is interesting but does not compare with biological evolution and the vast range of forms and structures that we see there. Our simulations could never generate such a range of structures because the creatures' representation, morphogenesis and mutation operators are fixed, even though the various probabilities of their application and effect may change. That is, although we have a general notion of the sorts of energy we are simulating, and this is encoded in our metamodel, we do not have a similar notion of a range of organisms. Therefore, the evolution we are exploring here is not fully open-ended. In order to do that we need a more abstract description of evolution.

## Conclusions

Our metamodel summarises the essential components of an energy-rich world which is a basic feature of real world evolution, and also of artificial life. We have shown that application of this metamodel in even a simple manner yields more complex, more interesting, results.

However, our experiments also make it obvious that we need much more in order to approach real open-ended evolution. In particular we must be able to modify both the creatures and the kinds of modifications that the creatures undergo. Our current simulations do not support this.

## Future Work

While interesting, our current simulation does not explore some aspects of the worlds implied by our metamodel. In

particular we have not explored the notion of entropy, which we believe should open up further different ways of creatures making their living. We have also not explored a non-homogeneous world with, for example, a range of different energy fluxes and different levels of friction which could make, again, different modes of existence feasible.

And, as we have discussed, we would like to investigate ways of extending the kinds of evolution that occur in order to more closely approach true open-ended evolution.

## Acknowledgments

The work described here is part of the CoSMoS<sup>3</sup> project, funded by EPSRC grant EP/E053505/1 and a Microsoft Research Europe PhD studentship.

## References

- [1] P. S. Andrews, F. A. C. Polack, A. T. Sampson, S. Stepney, and J. Timmis. The CoSMoS process, version 0.1: A process for the modelling and simulation of complex systems. Technical Report YCS-2010-453, Department of Computer Science, University of York, Mar. 2010.
- [2] C. Darwin. *On the Origin of Species*. John Murray, 1859.
- [3] P. S. di Fenizio. A less abstract artificial chemistry. In *Artificial Life VII*, pages 49–53. MIT Press, 2000.
- [4] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley, 1995.
- [5] S. Hickinbotham, E. Clark, S. Stepney, T. Clarke, A. Nellis, M. Pay, and P. Young. Diversity from a monoculture: effects of mutation-on-copy in a string-based artificial chemistry. In *ALife XII*, pages 24–31. MIT Press, 2010.
- [6] T. Hovard and S. Stepney. Environment orientation: an architecture for simulating complex systems. In *Proceedings of the 2009 Workshop on Complex Systems Modelling and Simulation*, pages 67–82. Luniver Press, 2009.
- [7] T. R. Malthus, R. Malthus, and G. Gilbert. *An Essay on the Principle of Population*. Oxford Paperbacks, 2008.
- [8] T. S. Ray. An Approach to the Synthesis of Life. In *Artificial Life II*, pages 371–408. Addison-Wesley, 1992.
- [9] T. J. Taylor. From artificial evolution to artificial life. Technical report, University of Edinburgh, 1999.
- [10] G. Turk. Sticky feet: Evolution in a multi-creature physical simulation. In *ALife XII*, pages 496–503. MIT Press, 2010.

<sup>3</sup><http://www.cosmos-research.org>.

# Autonomy of the internet: complexity of flow dynamics in a packet switching network

Takashi Ikegami<sup>1</sup>, Mizuki Oka<sup>2</sup>, Hirotake Abe<sup>3</sup>

<sup>1</sup> Graduate School of Arts and Sciences, The University of Tokyo, Japan

<sup>2</sup> Center for Knowledge Structuring, The University of Tokyo, Japan

<sup>3</sup> Cybermedia Center, Osaka University, Japan

## Abstract

Autonomy of the internet (web system) is studied by running an NS-2 simulator. A web system consists of three layers, they are the network, the transport and the application layer and an network simulator called NS-2 can simulate the transport layer of the web as a packet switching network (PSN). This paper reports on the complexity of mutually crossing packet flows which are comparable to other autonomous complex networks, such as real Hippocampus slices, Izhikevich neural networks, or the game of life. One unique feature common in all these systems is the coexistence of several synchronised patterns that we think of as the underlying mechanism of autonomy. In the case of PSNs, adaptive window sizes of each packet flow show synchronisation but only locally, and often chaotic behavior is displayed when congestion occurs. Also considering the packet flow in PSNs as gliders, this congestion allows gliders to bifurcate. We thus propose PSNs as a new experimental testbed for discussing the autonomy and adaptability of living systems.

## Introduction

Autonomy is one of the most important characteristics of living systems. Understanding this biological autonomy by reconstructing it using different media is one of the main purposes of Artificial Life studies. For example, the study of autonomous robots uses such an approach. A definition of an autonomous robot is its ability to achieve a task without people having to make commands. There are many examples such as Stefano Nolfi's 'garbage collectors' (Nolfi, 1997), Pfeifer's passive dynamic walker (Pfeifer et al., 2007), Honda's 'Asimo', Sony's 'dog robot' called Aibo, Kojima's 'Keepon' and so on. Some of these robots are "autonomously" detecting walls and avoiding cliffs in various ways. Self-charging robots have also been built already (e.g. a robot that uses snails for energy or a trilobite-like robot that monitors its own battery); so robots can become self-sustainable in that sense.

Rodney Brooks claimed that autonomous robots need not possess any representation of the environment but the environment itself is the representation. They explore the environment and solve a given task. This is a major feature of autonomous robots (Brooks, 1991). Such a concept of

autonomy still misses a very fundamental part of biological autonomy, as we are still easily able to distinguish between real and artificial creatures (Brooks, 2001).

A simple but primary definition of an autonomous system is a non-reaction system. For example, a fly's aviation is considered to be an autonomous behavior as it behaves independently from the environmental pattern (May et al., 2007; Takahashi et al., 2008). Another such autonomous dynamic is chaotic itinerancy (Ikegami, 2007); a high dimensional transition dynamic among pseudo attractors. Aoucuturier et al. (Aoucuturier et al., 2008) used this idea to create a dancing mobile robot. Besides a hard-shell robot, Hanczyc and Ikegami (Hanczyc et al., 2007; Hanczyc and Ikegami, 2010) studied a self-moving droplet. An oil droplet made of oleic acid and sized about 0.1 mm can move by itself and also react to environmental pH.

The underlying principle in all these examples is that an interaction between a system and its environment creates autonomy. In other words, a system can generate and maintain its own context which temporarily couples and decouples with the environmental context. More importantly, a system has its own dynamics without requiring an externally given task. A so-called 'default network' found in a brain's resting state is another example of such autonomy (Raichle et al., 2001). The definition of a default network is the brain activity observed while people are day-dreaming or doing non-specific tasks. A global (non-periodic) synchrony in neural activity was found to exist in the default network.

In this paper, we discuss the concept of autonomy using the example of web systems. Nowadays, web systems have become huge and complex enough to have consciousness-like states. Such web autonomy can be considered sufficiently close to biological autonomy. Corresponding to the non-periodic neural synchrony found in the default network, we will report the non-periodic behavior in a simulated web system.

In §2, we review the constitution of web systems, and in §3, we introduce an internet simulator called NS-2<sup>1</sup> which emulates the packet switching network (PSN) of the inter-

<sup>1</sup>The Network Simulator - NS-2: <http://www.isi.edu/nsnam/ns/>

systems	<i>brain</i>	<i>internet</i>	<i>ANN</i>	<i>the game of Life</i>
<b>basic element</b>	neuron & synapses	node & packet	coupled equation	2 states 2 dim. lattice
<b>structure</b>	small world	small world	random connection	regular lattice
<b>dynamics</b>	local/global synchrony	<b>this paper</b>	synchro/polychro	gliders/space ships
<b>memory</b>	semantic & episodic memory	Google DB & Twitter TL	attractors & CI	space pattern

Table 1: Comparison of five different network systems is be conducted; a real brain system, the internet, artificial neural nets (ANN) and the game of Life. The structure and dynamics of each network is depicted in the 2nd and 3rd row, respectively. Possible dynamics of the internet in terms of PSNs is discussed in this paper. In the 4th row, kinds of memory in each network is also described, where an ANN stores its memory in terms of attractors and chaotic itinerancy (CI) and the game of life stores its memory in terms of special spatial configurations.

net’s transport layer. In§4, data from the NS-2 simulation will be discussed with respect to dynamic stability. In §5, a simple question we can ask about the web autonomy such as ”what happens if everybody stops accessing the internet”, is examined. Finally, we will discuss what brings autonomy to a PSN.

### Web Systems

The internet has made great progress in the last 20 years and it has become a lifeline for human society. Its structure consists of roughly three layers; a network, a transport and an application layer. When studying the autonomous dynamics of the internet’s application layer, we can examine web crawlers and Google’s PageRank to see how the database is automatically organised and ranked. Many social network services (SNSs) such as Twitter are also worth noting. They mutually copy and reproduce personal timelines in massively parallel ways which is somehow complementary to what Google’s service is processing on their stored data.

On the other hand, what enables Google and Twitter to function correctly is a PSN on the transport layer and its backbone network layer. This creates a system that can be mutually connected on the internet with IP addresses on the network layer. The protocols used for communicating among those IP addresses are TCP or UDP. In particular, TCP is equipped with relatively intelligent software. Each network router sends a data flow by switching data packets. TCP plays an important role in delivering the data to the address without going missing nor permutation of packets. The sender controls the data amount and the router controls data routing.

The topological structure of the internet has been intensively studied and its small world property (Watts and Strogatz, 1998) is revealed. One property that a network has is a hub connection, and this is now widely known in generic information about transporting systems, e.g. gene networks or neural networks in the brain. A.L. Barabasi reported that such small world networks become even more robust when compared with random networks (Barabasi and Albert, 1999).

But we also think it is important to understand the flow dynamics on the internet rather than just its topological structure. Graham proposes PSNs as a new model for a brain system in place of a circuit switching network (Graham and Rockmore, 2011). Griffith et al. argue the similarity between Google’s PageRank system and how the mind works (Griffiths et al., 2007). These are the dynamic properties of a network and we hope that the minimal and prerequisite fundamental dynamics for a kind of intelligence and mind can be found in PSNs.

Indeed, the complexity of the internet’s dynamics has an equally curious property which we find in the human neural circuit. There have been several studies concerning dynamic complexity of PSN (see e.g. (Frommer et al., 2009)). The inherent complexity of PSNs can be seen at the level of producing consciousness-like macro phenomena, which Tononi and Edelman hypothesised with their concept of dynamic core and reentry (Edelman and Tononi, 2000).

We list characteristics in the Table 1 to compare PSNs with the other complex enough network systems. Neural synchronisation phenomena were discovered by Singer in the visual cortex of a cat (Singer and Gray, 1995). Such synchrony is also found as a self-moving pattern in Hippocampus slices (Takahashi et al., 2010) or in the massive number of artificial neural networks (ANN) (Izhikevich, 2000; Izhikevich and Edelman, 2008; Izhikevich, 2006). Here we only refer to the Izhikevich neural net, as this network is realistic in its scale and types of neural spiking. It should be noted that synchronisation is not always a global phenomenon but it is often observed as a local synchronisation or clustering of neural oscillation. In other words, different neural clustering in space and time can coexist. This is a universal phenomenon in generic coupled nonlinear systems (Kaneko, 1990).

What is more interesting is that a localized pattern can propel itself through space; we call these gliders and space-ships in the game of life. A glider or spaceshp is used to prove the universal computability of the game of life as demonstrated by William Poundstone (Poundstone, 1984). Indeed the role of a glider pattern is for a basic information packet to run through the system, and gliders sponta-

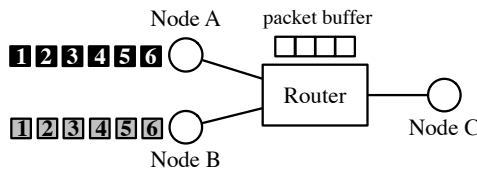


Figure 1: An example network has three nodes and one router. Nodes A and B send packets to node C creating two flows.

neously interact with each other to maintain the system’s autonomous information processing. One of the purposes of this paper is to look for similar phenomena in PSNs by taking packets as the simplest form of glider. This notion of autonomy is what we are going to seek in PSNs.

If a system is autonomous and sufficiently complex, we expect it to show various signs of intelligent behavior. One such intelligent behaviour is based on memory dynamics. Therefore, we put memory as the 4th row in the table. Here different kinds of memory are potentially stored in the networks. ANN stores memory as attractors (see e.g. Hopfield network (Hopfield, 1982)), which can be referred to as semantic memory, but it also stores episodic memory as chaotic itinerant dynamics of pseudo attractors (Nozawa, 1992; Tsuda, 2001; Tani, 1998). As discussed at the beginning of this section, the internet now mainly consists of two memory structures. One is Google’s Database (DB) and the other is Twitter’s time line (TL). We think these are related to semantic and episodic memory in real brain systems, except that they are about the application layer. However this is beyond the scope of this paper and will be reported in ASSC 15<sup>2</sup>.

Finally, the game of life stores memory in terms of special spatial configuration. The best known example of a cellular automaton’s (CA) memory might well be von Neumann’s self-reproducing automata (Neumann, 1967). Since the game of life can emulate any kinds of CA, we propose here that any powerful CA can become a universal Turing machine in the game of life.

## The Packet Switching Network Model

NS-2 is a simulator for a packet switching network (PSN). We claim that this network corresponds to the neural network of a brain system, where each connected neuron sends electric pulses to the others with different timing and strengths. At the end of this section we compare the basic properties of PSNs and neural networks, but first we explain how NS-2 works.

To illustrate how NS-2 works, let us consider a simple network where three nodes are connected through one router as depicted in Fig. 1. For example, when node A tries to

<sup>2</sup>ASSC 15 : The 15th annual meeting of the ASSC. <http://www.theassc.org/conferences/assc-15>

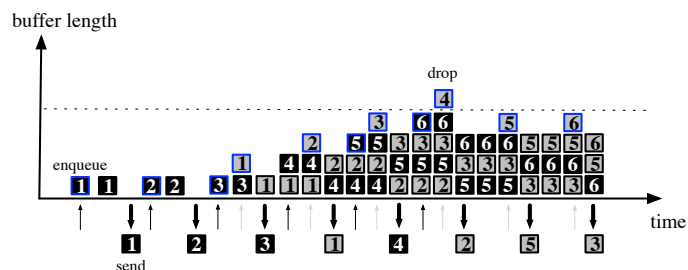


Figure 2: Changes occur in the buffer as the router receives and sends packets with the buffer size equal to four.

send a packet to node C, it goes through the router. The router has a certain length of packet buffer, say four. The packet will be sent to node C if the packet buffer is not over capacity. If the packet is over capacity, congestion occurs and the packet will simply be dropped. Fig. 2 shows the changes in the router’s buffer status when packets 1, 2, 3, 4, 5, and 6 from node A and B are sent to node C respectively. The figure shows the router’s buffer when two flows occur, one from node A to node C and the other from node B to C. The black arrow shows the arrival of the packet to the node, enqueueing the packet to the buffer, then sending the resulting dequeueing of the packet. These two events would show in the logged file of NS-2 as follows:

```
+ time A C 1 1
- time A C 1 1
```

where each line denotes ‘event’, ‘time’, ‘destination node id’, ‘arrival node id’, ‘flow id’ and ‘packet id’. The ‘+’ denotes the enqueueing event to the buffer and the ‘-’ denotes the dequeueing event to the buffer. Similarly, when the node arrives at either a route or a node, it will be logged in the file as:

```
r time A C 1 1
```

where ‘r’ denotes an arrival event. When the buffer becomes full and create a congestion, a dropping event occurs as shown in the figure for node B packet and it would be logged in the file as follows:

```
d time B C 2 4
```

When a drop event happens, that packet will always be lost. The Transmission Control Protocol (or TCP) is a mechanism designed to create more reliable transmissions. TCP sends a packet with a serial number. When a node receives a packet, it sends back the serial number which is called acknowledgement (or ACK). When the sender receives an ACK, then it sends the next packet. If the sender node does not receive an ACK for a certain period of time or receives ACKs with wrong sequence number, then it resends the same packet. However, as one can easily imagine, sending packets one by one is not efficient. To cope with this, TCP has a parameter called congestion window size which





Figure 3: Network topology for the experiment.

defines how many packets the sender can send at one time. This window size is advertised by the receiver node. To improve performance, the advertised window size needs to be set to 'large'. However, when the window size is too large, it creates congestion with other packets resulting in packet drops and consequent requests to re-send the packets.

While the advertised window size is imposed by the receiver, there is another window size imposed by the sender called the congestion window size, or called the "cwnd". When a new connection is established with a node, the cwnd is initialized to one segment (i.e. the segment size is announced at the other end). Each time an ACK is received, the cwnd is increased by one segment. The question about how to improve performance then becomes how to adjust the advertised window size and the cwnd size. The former is related to the amount of available buffer space at the receiver for the connection; the latter is based on the sender's assessment of perceived network congestion. It is important to note that the cwnd size continues to increase to a given threshold or until a drop event happens.

There are a number of different algorithms to increase the cwnd size. The one we used in this study is called Reno. The Reno algorithm increases the cwnd exponentially until the first packet drop occurs due to congestion. After the first drop, the cwnd is set to half then continues to update itself in a linear manner. When a drop happens, it again sets to half and starts to increase again and continues this process throughout.

As we have explained so far, PSNs (and the simulator, NS-2) have the following corresponding properties when compared to biological neural networks:

- Flow dynamics in PSNs correspond to the pulse trains of neural activities.
- A buffer size corresponds to the activation threshold of a neural firing. In the NS-2 model we use 10 as the buffer size and the threshold of a real neural cell is about 15 mV.
- Strength of the cwnd corresponds to synaptic strength. Here we have Reno algorithm to change the window size.
- A drop event corresponds to the fact that neural pulses cannot contribute to an overshooting event.

Having this correspondence in mind, we analyse and explore the PSN in the next section.

## Analysis

We have conducted experiments using NS-2 on a simple network topology with a 30 node setting where each node is

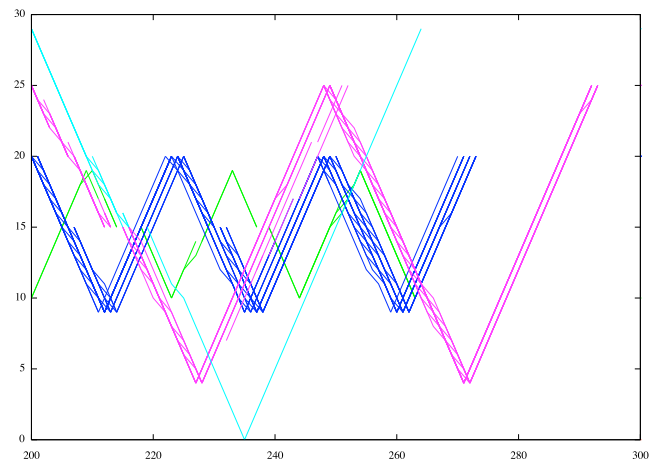


Figure 4: An example of spatio-temporal packet flow pattern. The horizontal axis is time (each step is 10 msec) and the vertical axis is the spatial node (here the total number of nodes is 30). As this figure shows, each packet flow spontaneously bifurcates so that lines are multiplied. Every flow shows concatenated "V-shaped" pattern, since every successfully received packet is followed by ACK signal sent back to the sender.

connected to the next node as depicted in Fig. 3. The analysis will be on the flow dynamics, the congestion phenomena and the robustness of the flow patterns when pouring a temporary flow from outside. We will explain these below.

## Flow dynamics

A unique characteristic of a PSN is a self-tuning cwnd size for each flow in the network. In the first simulation, we created 30 flows in which each router sends a series of packet data to its neighbors through an optimised routing pathway (or trace). All flows are set to have an equal length. As described above, each packet between connected nodes ( $i \rightarrow j$ ) is characterised by a triplet (+,-,r) state, where the state "+" corresponds to "the packet in node  $i$  is ready to send", "-" to "the packet has been sent to node  $j$ " and "r" to "the packet has been received by node  $j$ ". Using this information, we can visualise the spatio-temporal flow pattern as shown in Fig. 4.

As for basic observations, we see i) The more numbers of nodes the flow travels, the more transport time is required; ii) Due to spontaneous time delays, packets that constitute the same flow arrive in different timeframes, which causes the bifurcation of flow pathways. This bifurcation pattern can be different for each flow; iii) Even within the same flow and in between the same traces, the bifurcation pattern can vary temporarily.

It should be noted that the bifurcation of flow path due to the time delay in point (ii) above is a novel feature in dy-



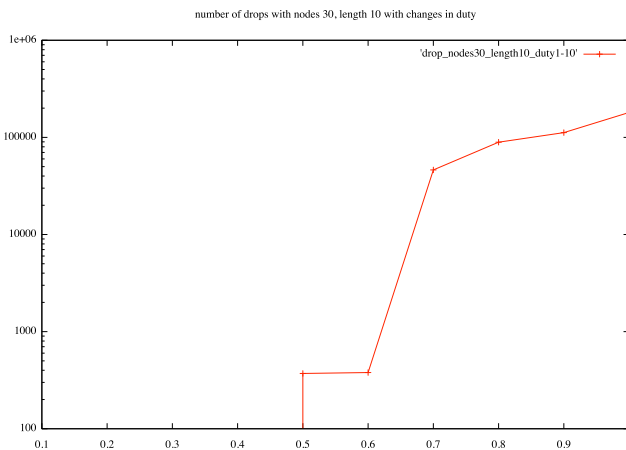


Figure 5: The number of drop events as a function of the integrated input packets (we call a 'duty ratio') on each source node. Input packets are given periodically for each node. When duty ratio =  $x (< 1)$ , it means that each source node sends packets for  $x$  seconds and rests for the next  $(1 - x)$  second for every cycle.

dynamic systems. In PSNs, flow is spontaneously quantized into a series of packets when transporting to other nodes. This clustering event is not written in the form of an "equation" in the PSN but happens only as a result of congestion and timing. Although such spontaneous clustering is similar to congestion patterns studied in traffic models (Chowdhury et al., 2004), PSNs have drop events and ACK signals. In the case of traffic jams, vehicles or ants will never disappear. This traffic jam phenomenon is called congestion. Bifurcation of flow pattern is correlated with this congestion pattern, which we will focus on in the next section.

### Congestion Flow

As explained in the previous section, the source node of each flow tunes the buffer size and the cwnd size to reduce the drop events. When the amount of flow becomes larger than a specified volume, congestion occurs spontaneously and the number of drop events increases exponentially as the amount of flow increases. Fig. 5 shows an increase in the number of drops as the ratio of the packet flow period to the frame increases.

The drop events trigger the clustering of the window size. In the first hundreds steps, each window size is mutually tuned and their phases are synchronised as shown in Fig. 6. This is known as TCP global synchronisation<sup>3</sup>. In the figure, all flows are set to have an equal length. In this case, even though the cwnd size changes from a periodic to an aperiodic state the packet flow is mostly periodic. Because

<sup>3</sup>TCP Global Synchronisation : [http://en.wikipedia.org/wiki/TCP\\_global\\_synchronization](http://en.wikipedia.org/wiki/TCP_global_synchronization)

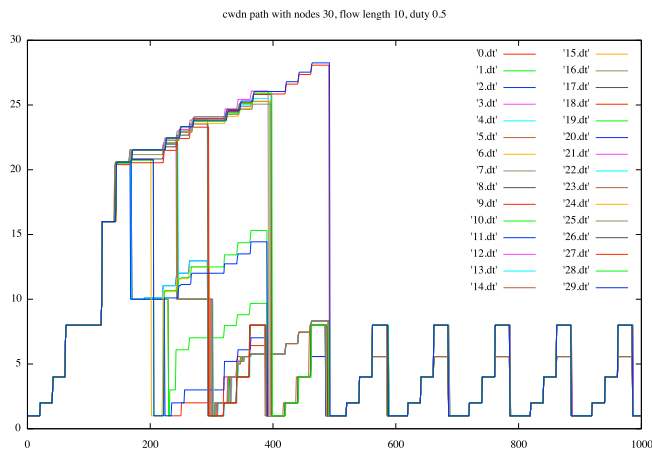


Figure 6: An example of the cwnd dynamics. Each window size of the flow is overlaid multiple times. Here the system has a few drop events so that the network settles down to a periodic synchronized pattern after four seconds.

almost all the drop events occur at the source of the flow, the drop events change the cwnd dynamics but not the packet flow pattern.

We artificially create a special topology that produces massive congestion in the middle nodes (i.e. between nodes 14 and 15 of the 30 nodes). In this case, both the flow patterns and the cwnd dynamics become unstable, as the drop events occur not only at the source but also at the relay nodes. Some examples of the flow patterns and cwnd patterns are depicted in Fig. 8. The transport time of every flow shows a power law behavior of the exponent being equal to -2 as shown in Fig. 7. The connection between nodes 14 and 15 becomes a bottleneck and determines the entire time scale.

When a cwnd dynamic settles into a periodic state, its periodicity becomes almost consistent with its varying window size. In the case of aperiodic cwnd dynamics shown in Fig. 8, we classified this into five clusters based on the temporal oscillating pattern as we do for the dynamical systems.

1. Periodic state: The window size changes periodically in a stepped way. Fig. 8-(a) represents this cluster.
2. Chaotic state: The window size changes in an aperiodic way. In the case of Fig. 8-(b,c), we have two different chaotic behaviours; one with fast amplitudes varying in time and one with slow amplitudes changes in time, where their time scales also show some variations.
3. Intermittent chaotic state: The periodic oscillation of window size is intermittently perturbed by a burst of large window size. The other intermittent behaviour is that the amplitude almost periodically oscillates around a certain

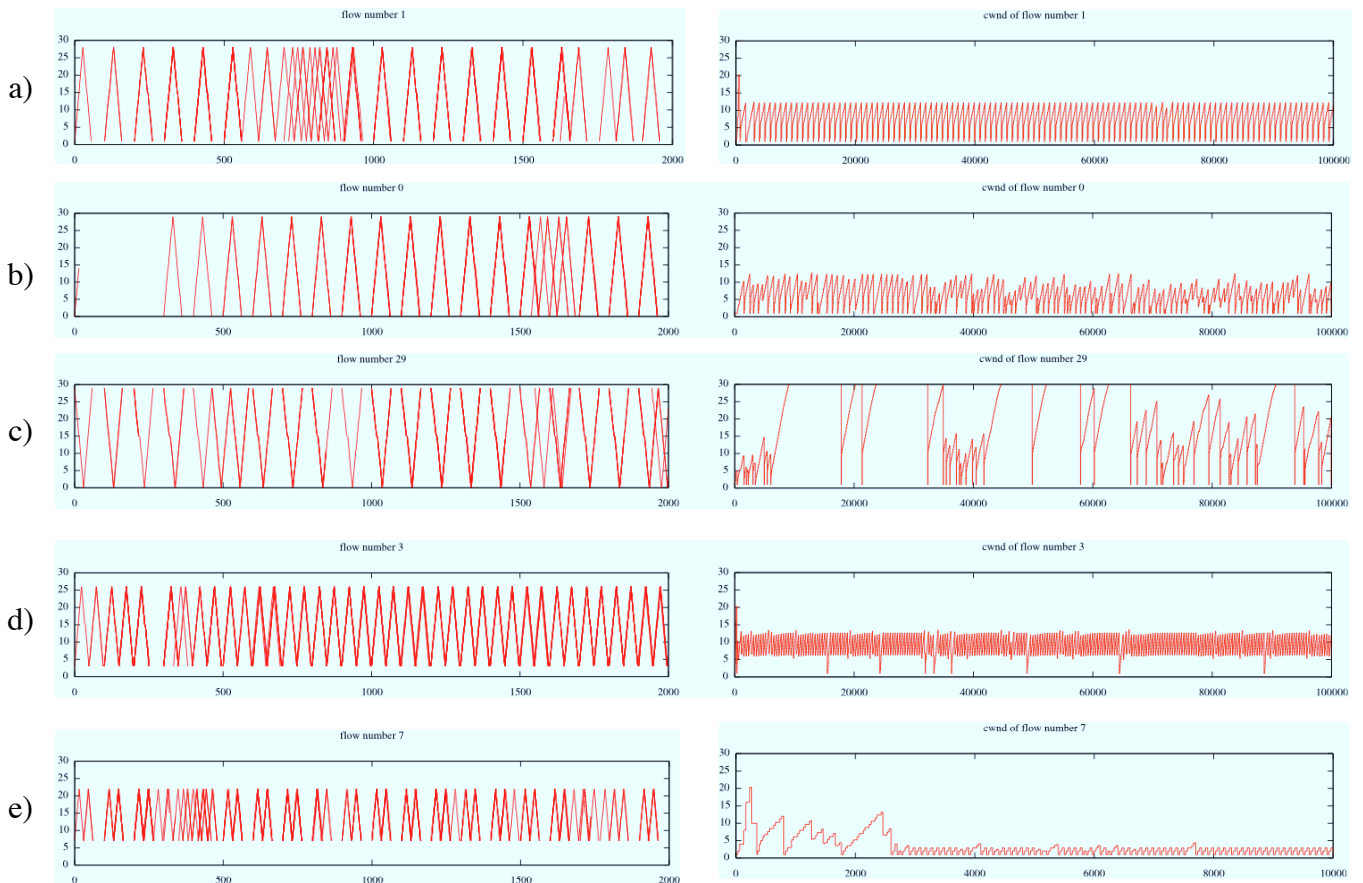


Figure 8: Examples of five categories of cwnd dynamics (right) and the corresponding packet flow pattern (left). From the above, these are: a) periodic, b) chaotic type 1, c) chaotic type 2, d) intermittent chaos type 1, and e) intermittent chaos type 2. Time scale is set from 0 to 100,000 except for the case e), since the oscillation of case e) is much faster compared to other cases. See the details in the text.

value but is intermittently perturbed by a larger or smaller (often null) window size. Both of these can be observed in Fig. 8-(d,e).

Flows synchronised in the same clustering pattern can be found in the spatial neighbors with some exceptions. It should be noted that the chaotic synchrony is what we compare with the Hippocampus slice or Izhikevich neural ensemble as a candidate for the origin of autonomy and a computation primitive. As discussed in §2, these synchronised patterns are important in maintaining the functionality of the network as a whole. In particular, we propose that these synchronised patterns may be a source of PSN autonomy.

### Perturbation

Let us perturb the flow network by pouring an extra flow from outside at a certain time duration. A stable network, where both flow and cwnd pattern become periodic, will remain robust against the perturbation, i.e. the flow pattern and cwnd dynamics will remain periodic. On the other hand, a

network that has massive congestion at the middle point is less robust against perturbation. Comparisons before and after perturbation demonstrate that flow pattern (Fig. 9) and cwnd dynamics will be different. In other words, the flow state can be said to have chaotic instability as it amplifies the small difference caused by the extra flow input.

It can be said that for this special network, the flow state becomes less robust against perturbation. But we also interpret the state as adaptive because it never falls to a fixed flow state.

### Discussion

The autonomy we are looking for is having the flexible internal dynamics to change responses against external inputs. A certain amount of chaotic dynamics may be responsible for this. In previous studies, we have partially proven that an autonomous robot equipped with the coupled FitzHugh-Nagumo equations shows such autonomy (Aucouturier et al., 2008). Analysis of how such a robot can

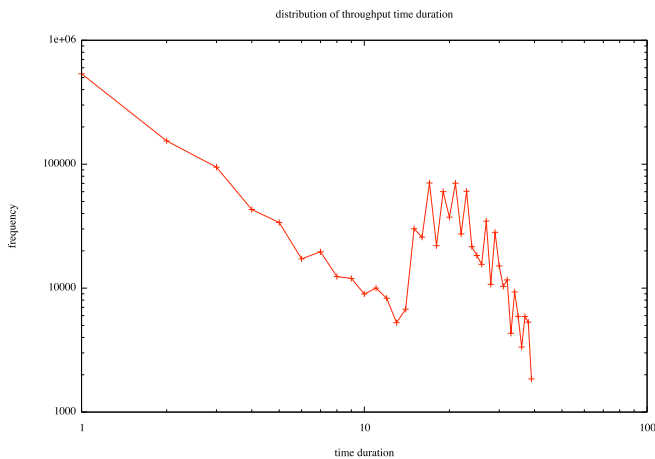


Figure 7: Distribution of throughput time duration. The distribution of the shorter time duration shows the power law behavior, which corresponds to the bottleneck connection between the 14th and 15th nodes.

interact with the environment was the focal point.

In the present paper, we have studied PSNs to reveal the internal flow dynamics and the system responses to external pulse inputs. When increasing the amount of flow from the outside, we showed that cwnd dynamics change from periodic to chaotic. In the case of a network with a bottleneck edge, the transport time of each flow obeys the power law and the real packet flow becomes chaotic for a long period of time.

In §1, we posed a question, "what happens if everybody stops accessing the internet?". An answer to this question might be "it won't stop immediately but will last a long time because it does not attain a stable pattern as shown from the PSN experiment". The ever-changing nature of chaotic and intermittent clustering may drive the autonomy of the internet even with periodic inputs supplied, for example, by automated web crawlers. If a simple one-dimensional PSN can have complex clustering patterns, the internet with its massive data flow should have ever-lasting and changing clustering, thus making it autonomous. We believe these findings correspond to the examples of complex networks in Table 1. That is, autonomous networks can develop complex local/global space time clustering or gliders.

We also claim that PSNs are a novel class of dynamic systems that spontaneously bifurcate their flow structure and may be the backbone of the internet today. The corresponding gliders in the game of life and other intelligent systems in Table 1 remain stable. However in the case of PSNs, those localised patterns can bifurcate. This bifurcation of glider-like patterns is why we think this PSN is a new interesting testbed for Artificial Life studies. As for future investigations, we have to pay more attention to other novel features

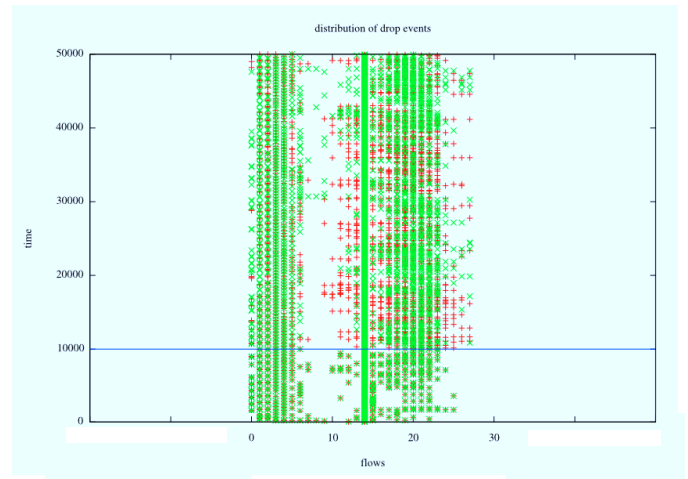


Figure 9: Comparison of space time plots of drop events for the original and the perturbed network (the horizontal axis denotes the network node IDs and the vertical axis denotes time steps). The perturbation is introduced as a pulse packet flow of duty ratio=1 poured during 10,000 and 20,000 msec-onds. After the perturbation, a network does not come back to the original state.

of PSNs. For example, dynamic routing and another TCP/IP will be a future research target.

Our analysis here is about the transport layer not the application layer. Examples of autonomous software in the application layer include web crawlers, peer-to-peer software, and SNS bots. The existence of those two layers does contribute to making a more complex autonomous system. Within a computer system, an example of a software algorithm that generates chaotic dynamics in the hardware layer was reported by Berry et al. in 2006 (Berry et al., 2006). We are now studying the autonomous behavior in the application layer using the notion of clustering dynamics found in the PSN reported in this paper.

## References

- Aucouturier, J.-J., Ogai, Y., and Ikegami, T. (2008). Using chaos to trade-off synchronization and autonomy in a dancing robot. *Trends and Controversies, IEEE Intelligent Systems*, 2:74–85.
- Barabasi, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286:509–512.
- Berry, H., Perez, D. G., and Temam, O. (2006). Chaos in computer performance. *Chaos*, 16:01311011–013110115.
- Brooks, R. (2001). The relationship between matter and life. *Nature*, 409:409–411.
- Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence Journal*, 47:139–159.
- Chowdhury, D., Nishinari, K., and Schadschneider, A. (2004). Self-organized patterns and traffic flow in colonies of organisms. *Phase Transitions*, 77:601–624.

- Edelman, G. and Tononi, G. (2000). *A Universe of Consciousness: How Matter Becomes Imagination*. Basic Books.
- Frommer, I., Harder, E., Hunt, B., Lance, R., Ott, E., and Yorke, J. (2009). Bifurcations and chaos in a periodically probed computer network. *International Journal of Bifurcation and Chaos*, 19:3129–3141.
- Graham, D. and Rockmore, D. (2011). The packet switching brain. *J. Cogn. Neurosci.*, 23(2):267–76.
- Griffiths, T. L., Steyvers, M., and Firl, A. (2007). Google and the mind: Predicting fluency with pagerank. *Psychological Science*, 18:1069–1076.
- Hanczyc, M. M. and Ikegami, T. (2010). Chemical basis for minimal cognition. *Artificial Life*, 16(3):233–243.
- Hanczyc, M. M., Toyota, T., Ikegami, T., Packard, N., and Sugawara, T. (2007). Chemistry at the oil-water interface: Self-propelled oil droplets. *J. Am. Chem. Soc.*, 129(30):9386–9391.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *J. Proc Natl.Acad. Sci.*, 79:2554–2558.
- Ikegami, T. (2007). Simulating active perception and mental imagery with embodied chaotic itinerancy. *J.Consciousness Studies*, 14:111–125.
- Izhikevich, E. M. (2000). Neural excitability, spiking, and bursting. *International Journal of Bifurcation and Chaos*, 10:1171–1266.
- Izhikevich, E. M. (2006). Polychronization: computation with spikes. *Neural Comput.*, 18:245–282.
- Izhikevich, E. M. and Edelman, G. M. (2008). Large-scale model of mammalian thalamocortical systems. *PNAS*, 105:3593–3598.
- Kaneko, K. (1990). Clustering, coding, switching, hierarchical ordering, and control in network of chaotic elements. *Physica D*, 41:137–172.
- May, A., Hsieh, C., Sugihara, G., and Brembs, B. (2007). Order in spontaneous motion. *PlosOne*, 5:443.
- Neumann, J. V. (1967). *Theory of Self-reproducing Automata* (ed. Arthur W. Burks). University of Illinois Press.
- Nolfi, S. (1997). Evolving non-trivial behaviors on real robots: A garbage collecting robot. *Robotics and Autonomous Systems*, 22:187–198.
- Nozawa, H. (1992). A neural network model as a globally coupled map and applications based on chaos. *Chaos*, 2:377–386.
- Pfeifer, R., Lungarella, M., and Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *Science*, 318:1088–1093.
- Poundstone, W. (1984). *The Recursive Universe: Cosmic Complexity and the Limits of Scientific Knowledge*. William Morrow & Co.
- Raichle, M. E., MacLeod, A. M., Snyder, A. Z., Powers, W. J., Gusnard, D. A., and Shulman, G. L. (2001). Inaugural article: A default mode of brain function. *PNAS*, 98:676–82.
- Singer, W. and Gray, C. M. (1995). Visual feature integration and the temporal correlation hypothesis. *Ann Rev Neurosci*, 18:555–586.
- Takahashi, H., Horibe, N., Shimada, M., and Ikegami, T. (2008). Analyzing the house fly’s exploratory behavior with autoregression methods. *J. Phys. Soc.*, 77:084802.
- Takahashi, N., Sasaki, T., Matsumoto, W., Matsuki, N., and Ikegaya, Y. (2010). Circuit topology for synchronizing neurons in spontaneously active networks. *PNAS*, 107:10244–10249.
- Tani, J. (1998). An interpretation of the ‘self’ from the dynamical systems perspective: Aconstructivist approach. *Journal of Consciousness Studies*, 5:516–42.
- Tsuda, I. (2001). Toward an interpretation of dynamic neural activity in terms of chaotic dynamical systems. *Behav. Brain Sci.*, 24:575–628.
- Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of small-world networks. *Nature*, 393:440–442.

# Simulation of cell-like self-replication phenomenon in a two-dimensional hybrid cellular automata model

Takeshi Ishida<sup>1</sup>

<sup>1</sup> Nippon Institute of Technology  
ishida06@ecoinfo.jp

## Abstract

An understanding of the generalized mechanism of self-reproduction is fundamental to applications in various fields, such as the mass-production of molecular machines in nanotechnology. We have developed a model for the simulation of cellular self-reproduction in a two-dimensional cellular automaton, and we have demonstrated that the following three functions can be realized: (1) formation of a border similar to a cell membrane, (2) self-replication while maintaining carrier-containing information, and (3) division of the cell membrane while maintaining the total structure. Furthermore, we have constructed a hybrid cellular automaton model. To reduce the number of transition rules, we considered not only the state transition rules but also the concentration diffusion in the Gray Scott model, in which the self-reproduction phenomenon emerges under certain parameters.

## Introduction

An understanding of the generalized mechanism of self-reproduction is considered fundamental to applications in various fields, such as the mass-production of molecular machines in nanotechnology and artificial synthetics in biology (synthetic biology). Furthermore, it is difficult to construct large, complex machine systems that exceed a certain size, using a top-down approach. Therefore, such complex systems must be constructed using a bottom-up approach based on the phenomenon of biological self-organization. Thus, it is crucial to elucidate not only the details of real cellular reaction networks but also the conditions necessary for self-organized and self-replicating cells.

A system that can simulate the self-reproduction of a cell must fulfill the following requirements. 1) It can express phenomena of nanolevel molecular behavior such as the Brownian movement. 2) It can express a chemical reaction system. 3) It can express the shape (difference in reaction process according to the shape) of compounds such as proteins. 4) It can express the emergence of macro shape and function for a bottom-up approach. For such a calculation, a particle system model is a potentially superior option.

Fifty years ago, von Neumann (1966) initiated a study on self-reproduction models from a mathematical viewpoint. His study theoretically proved the possibility of constructing a self-reproducing machine using cell states and the transition rules of two-dimensional square cells. However, von Neumann's

self-reproducing machine was large; therefore, it is difficult to implement this machine perfectly in a computer system (Mange, 2004). In 2010, Hutton (2010) implemented and simulated over its entire replication cycles. Later, Langton (1989) developed a simple machine capable of self-reproduction, by abandoning the completeness of von Neumann's machine; although its shape was quite simple and it could reproduce specific shapes, the rules of transition were complicated. The derivation of transition rules using genetic algorithms has been investigated (Reggia, 1998)(Sipper, 1998); however, it is difficult to derive the generalized rules.

Historically, researchers have attempted to develop a mathematical model to simulate the morphosis of living matter. Studies on the reproductive models of a body surface design, namely, the Turing model (Turing, 1952), and those on the leaf vein pattern of a plant (Feugier, 2005) and mollusk shell patterns (Meinhardt, 2003) are examples of previous research. In addition, many researchers have used a cellular automaton model to study tissue or tumor growth. Although these models can simulate a number of features of biological self-reproduction on a computer, they cannot reproduce the entire body on the basis of unified equations and rules, such as cytodifferentiation by gene expression→morphosis of cells→organogenesis→emergence of function.

In our previous study (Ishida, 2010) we developed a model for the simulation of cellular self-reproduction in a two-dimensional cellular automaton. We demonstrated that the following three functions could be realized by the transition between two adjacent cells.

- (1) Formation of a border similar to a cell membrane.
- (2) Self-replication while maintaining carrier-containing information (information carrier).
- (3) Division of the cell membrane while maintaining the total structure of the cell.

In this study, we demonstrate the self-reproducing ability of a shape that is similar to that of a real living cell. Figure 1 shows the results of a cell-type self-reproducing two-dimensional cellular automaton. It is important to note that the objective of this study is not to clarify all the necessary and sufficient conditions for self-reproduction. Instead, we consider the possibility of simulating self-replication in a real dynamic chemical reaction environment by applying the transition rules determined in this study. A similar previous studies by Ono, Ikegami (2000), and Hutton (2007). Ono & Ikegami does not completely lead to the replication of the



same cell. Hutton's work involves self-reproduction that does not include information carriers such as genes. The latter point indicates the novelty of the present study.

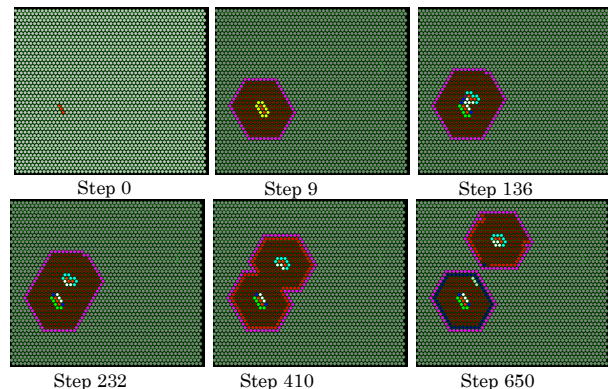


Figure 1 Results of a cell-type self-reproducing two-dimensional cellular automaton (Ishida, 2010)

## Objective

In this study, we constructed a hybrid cellular automaton (CA) model. Figure 2 shows the outline of the hybrid model. To reduce the number of transition rules, we considered not only the state transition rules but also the concentration diffusion of the field. We chose the Gray Scott model (GS model) (Gray, 1984), in which the self-reproduction phenomenon emerges with certain parameters. In this hybrid model, information carriers trigger the self-reproduction phenomenon of the GS model, and a cell membrane is formed by a part of the specific concentration of the GS model. If a single cell is being simulated, cell membrane formation is possible using a linear diffusion equation. This is difficult to accomplish using a simple linear model, and the GS model is necessary to fill the space while multiple cells are adjacent to one another and to maintain the distance between them.

This model is new, and it can be combined with existing models, such as the reaction diffusion equation models in the CA model. We express a macromolecule system in the CA model, and we express the small molecule-based reaction system that constitutes a reaction diffusion model, because the calculations become enormous when we calculate the reactions of all the molecules.

The simulation of a real living cell was considered difficult to express with only two phases, but it was based on future development. Furthermore, because a simple chemical reaction system can be substituted for the GS model, it is thought that we can simplify the model in the future.

As shown in Figure 3, we arranged the transition rules in the CA model and the GS equation parameters in two-dimensional space in order to simulate the duplication of hereditary information carriers, the encapsulation of information carriers by a cell membrane, and maintenance of the shape of the membrane.

Cellular automata possess characteristics that can help us understand the association between transition rules and results so that a state is determined solely by the rules governing the

transition between adjacent cells. In addition, the cellular automaton model has already been applied to discrete particle simulation, as in the case of fluids. For these reasons, it is theoretically possible to apply the transition rules to chemical or particle-collision systems.

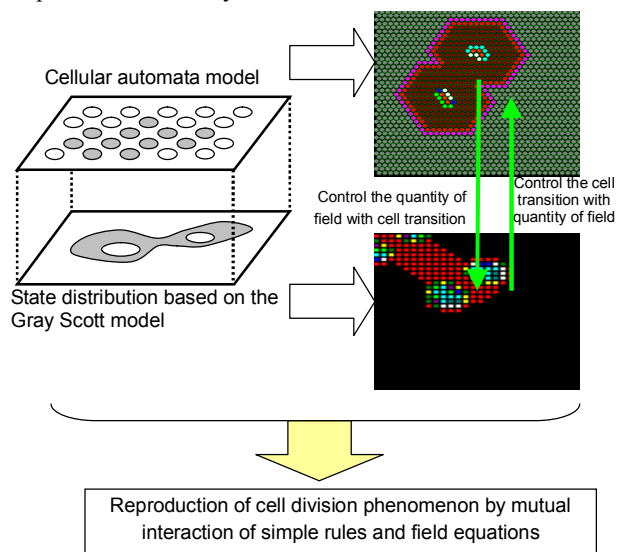


Figure 2 Outline of hybrid cellular automaton model

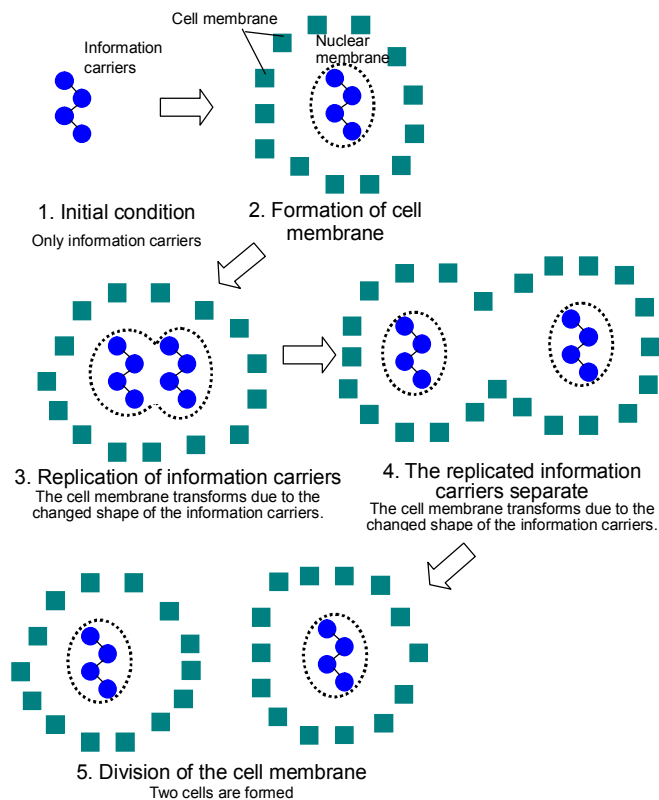


Figure 3 Conceptual diagram of cell-type self-reproduction in two-dimensional cellular automata

## Research Method

### Cellular automaton model

A two-dimensional hexagonal grid model was used in this study (Figure 4). Although square grids are typically used for two-dimensional cellular automata, a hexagonal grid model was used in this study for two reasons.

- (1) In the case of a square grid, the state of an automaton in the next step is determined on the basis of the state of the cell itself and the states of the eight adjacent sites. This increases the number of transition rules, and consequently their complexity. In the case of a hexagonal grid, the state of the next step is determined by the state of the cell itself and the states of the six adjacent sites. This reduces the number of transition rules.
- (2) Isotropy in the horizontal/vertical and diagonal directions is maintained in a hexagonal grid but not in a square grid.

The cell automaton was constructed according to the transition rules so that the state of the next step was determined by the state of the cell itself and the states of the six neighboring cells. Each cell had a state (0–19 states) and a direction (6 directions) as attributes.

0: State of non-being

1: States in which hereditary information carriers (Only states 1 and 2) have a directional attribute (any one of the 6 directions)). We can describe various types of information by creating subspecies (1-a, 1-b, etc.) for state 1.

2–10: States in which the nuclear membrane surrounds the hereditary information carriers

11–18: States that constitute space within the cell

19: States that consist of the cell membrane

In a hexagonal grid, the calculations start from a certain initial condition. As shown in Figure 5, the transition rules were divided into the following 4 phases: 1) state transition as regards cell membrane formation, 2) division of the information carriers, 3) movement of the information carriers, and 4) formation of a nuclear membrane surrounding the information carriers. In other words, we first applied the transition rules for cell membrane formation and settled the total states in all cells. Then, we applied the transition rules for the division of information carriers, after which we applied the transition rules for movement of the information carriers and formation of the nuclear membrane.

To induce objective state transitions of the cellular automata, we added transition rules to remove the unnecessary side effect reaction at the same time. We divided the

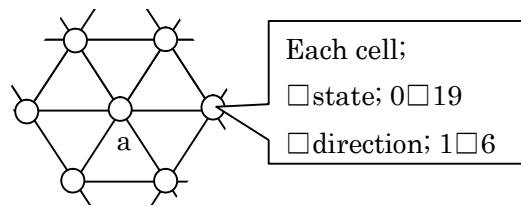


Figure 4 Hexagonal grid model used in this study

transitions into 4 phases to discover the transition rules, because discovery of the entire set of transition rules was difficult to achieve all at once.

### Gray Scott (GS) model

The cell transition patterns in this cellular automaton model resemble those of physical phenomena. Thus, we considered the possibility of replacing the transition rules with those of a non-linear quantity model such as the GS model. The equations for the GS model are given below. The self-replication patterns occur under certain conditions ( $D_u = 0.04$ ,  $D_v = 0.02$ ,  $F = 0.02$ , and  $k = 0.06$  in this study).

The initial concentrations of  $U$  and  $V$  assumed for the differential equation of the GS model were 1.0 and 0.0, respectively. This is a steady state in which there is no change in the concentration distribution. When there is a change in the concentration level in some spatial position, this triggers a dynamic change. When state 19 exists, the concentration distribution in the GS model in the same spatial position changes (from  $U = 1.0$  and  $V = 0.0$ , to  $U = 0.25$  and  $V = 0.35$ ). This unstable state leads to changes in the concentration distribution of the GS model.

On the other hand, as regards the action from the GS model to CA model, it is as follows. We calculated the ratio  $(500-U)/V$  of density  $U, V$  in the GS model and divided it into 10 parts between the minimums and the maximum of the value, and thus derived a potential level (1-10). A transition was induced in the CA model space when the condition appeared of a potential level shown in Table 4 on the GS model.

In addition, in the GS model, we can clarify the parameter set for when a self-reproduction design appears, but we cannot control the size of the self-reproduction design. Therefore, we adjusted the space scale of the CA model and the space scale of the GS model so that a cell membrane that encapsulated an information carrier was formed. The theoretical determination of the space scale method requires further investigation examination.

$$\frac{\partial V}{\partial t} = D_v \Delta V - U^2 V + F(1 - V) \quad (1)$$

$$\frac{\partial U}{\partial t} = D_u \Delta U + U^2 V - (F + k)U \quad (2)$$

### Transition rules

Each cell was renewed by the transition rules, and the state of the next step was determined by the state of the cell and the states of its 6 neighboring sites. The transition rules are presented in Tables 1–4. We have not yet discovered a method with which to derive transition rules automatically according to a uniform law. Therefore, we constructed transition rules step-by-step according to the movements of the automaton.

In this hybrid model, the information carriers first activate the GS model. Cell membrane states appear under certain concentrations of the GS fields. The movement of the nuclear membrane was controlled by the concentration of the GS fields.



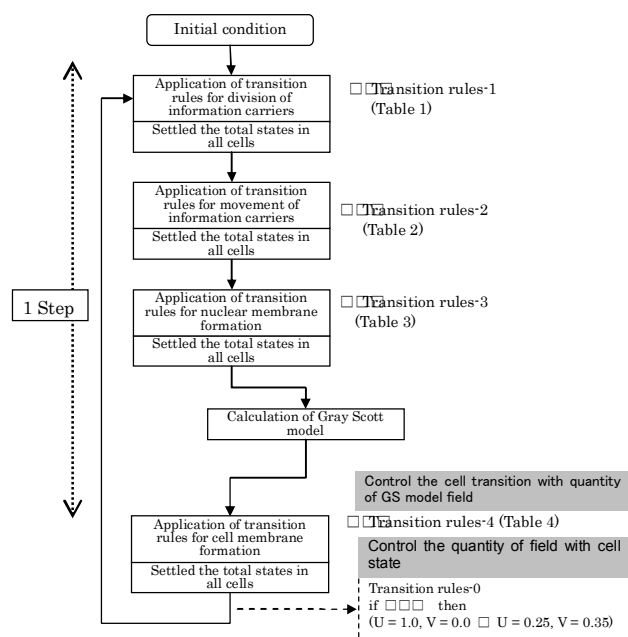


Figure 5 Calculation flow

## Initial conditions

Figure 6 shows the initial condition. The entire cell is in state 19. State 1 indicates the information carriers, three of which are arranged consecutively in the central part. The

intersection of state 19 and state 1 triggers the GS calculation. The purpose of this study is to find a minimum set of transition rules to achieve self-reproduction in a two-dimensional cellular automaton space. Our transition rule does not realize self-reproduction in any initial state.

## Results

Figure 7 shows the process of cell membrane formation and the process of the division of information carriers within the cell membrane. We carried out calculations for 101 steps; some of our results are shown in the figure. In each image in the figure, the upper part is the CA model and the lower part is the distribution of the potential level by the GS model. In this way, we were able to replicate the phenomenon of cell-like division.

Table 5 shows the number of transition rules for the cellular automaton model (Ishida, 2010) and the hybrid model. Using the hybrid model, we reduced the number of transition rules. In the case of the CA model, the transition rules to synchronize the cell-centered nuclear shape and the shape of the cell membrane were complicated. In the hybrid model, on the other hand, self-replication was possible with fewer rules, such that a cell membrane was formed on areas of a specific concentration. As compared with the CA model, the hybrid model is complicated in terms the calculation of the GS model; however, simpler rule description will be possible in the future because the GS model can replace the simple metabolism system.

Table 1 Transition Rule 1 (division of the information carriers)

	State	Direction	Transition of central cell (state)	Transition of central cell (direction)	Supplementary explanation	
1	3300311	11	2	5		
2	3300211	510	2	5		
3	3300211	511	2	5		
4	3303211	510	2	5		
5	3303211	511	2	5		
6	3003213	500	2	5		
7	3300322	55	1	6		
8	3303122	655	1	6		
9	3003123	650	1	6		
10	2213311	5560011	4	0		
11	2211411	5566011	4	0		
12	2211411	5566010	4	0		
13	2311413	5066000	4	0		
14	1133344	6600000	1	1		
15	1133144	6600100	1	1		
16	1333143	6000100	1	0		

Supplementary explanation of Table 1 and Table 2

State

--	--	--	--	--	--	--

Direction

--	--	--	--	--	--	--

State and direction of each cell of progression in order of a - g from a central cell.

Table 2 Transition Rule 2 (movement of the information carriers)

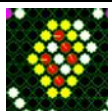

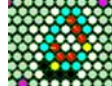



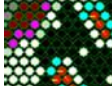


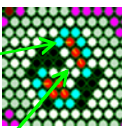
	State	Direction	Transition of central cell (state)	Transition of central cell (direction)	Supplementary explanation		
1	3300031	1	4	0	Control of DNA division	        	
2	8800041	1	4	0	movement of terminal		
3	8800411	11	1	1			
4	1114844	1110000	4	0	Movement of middle cell		
5	1114444	1110000	4	0			
6	8808111	111	1	1	Movement of middle cell( in front of tip)		
7	1111444	1111000	4	0			
8	8808111	110	1	1	Movement of tip		
9	1111444	1011000	4	0			
10	8008116	100	1	0			
11	1611445	1000	4	0			
12	8008118	100	1	0			
13	1811446	1000	4	0	continual movement of terminal		
14	1811448	1000	4	0			
15	1118844	1110000	4	0			
16	8611800	1000	1	0			Movement of tip
17	1544116	100	4	0			
18	8811800	1000	1	0			
19	1644118	100	4	0			
20	1844118	100	4	0	Movement of middle cell		
21	8111808	11000	1	1			
22	1444111	1000110	4	0	movement of terminal		
23	8118008	110000	1	1			
24	1445811	1000011	4	0			
25	1448811	1000011	4	0			

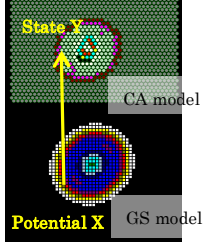
Table 3 Transition Rule 3 (formation of the nuclear membrane surrounding the information carriers)

	Central Cell	Conditions of six neighborhoods	Transition of central cell (state)	Supplementary explanation	
1	$\neq \textcircled{1}, \textcircled{2}, \textcircled{4} \sim \textcircled{9}$	$\textcircled{1} \geq 1$	$\textcircled{3}$	formation of the nuclear membrane	
2	$\neq \textcircled{1}, \textcircled{2}, \textcircled{4} \sim \textcircled{9}$	$\textcircled{2} \geq 1$	$\textcircled{3}$	"	
3	$\textcircled{3}$ and (Potential Value = 7)	-	$\textcircled{8}$	"	
4	$\textcircled{3}$	$\textcircled{8} \geq 1$	$\textcircled{8}$	"	
5	$\textcircled{4} \sim \textcircled{9}$	$(\textcircled{1} < 1) \text{ and } (\textcircled{8} \geq 1)$	$\textcircled{0}$	nonessential removal between information	

•The circled number the state of the cell. (ex.  $\textcircled{0}$  indicates state 0,  $\textcircled{1}$  indicates state 1)

•Method of describing condition : e.g., " $\textcircled{1} \geq 1$ " indicates that there is more than one cell in state 1 among the six neighborhoods.

Table 4 Transition Rule 4 (formation of the cell membrane)

	Central Cell	Conditions of six neighborhoods	Transition of central cell (state)	Supplementary explanation	
1	Potential Value = 1 or 2	-	$\textcircled{16}$	formation of the cytoplasm	
2	Potential Value = 3 or 4	-	$\textcircled{17}$	formation of the cytoplasm	
3	Potential Value = 5	-	$\textcircled{19}$	formation of the cell	
4	Potential Value = 6	-	$\textcircled{14}$	formation of the cytoplasm	
5	Potential Value = 7	-	$\textcircled{13}$	formation of the cytoplasm	
6	Potential Value = 8	-	$\textcircled{0}$	formation of the constitutive space in the cell	
7	Potential Value = 9	-	$\textcircled{0}$	formation of the constitutive space in the cell	
8	Potential Value = 10	-	$\textcircled{13}$	formation of the cytoplasm	

•The circled number show the state of the cell. (ex.  $\textcircled{0}$  indicates state 0,  $\textcircled{1}$  indicates state 1)

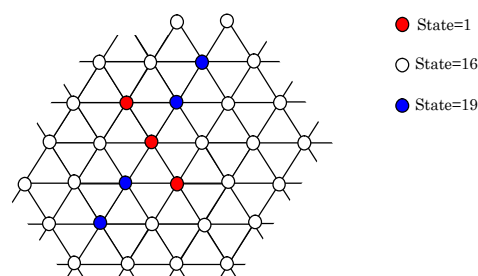


Figure 6 Initial conditions

Table 5 Number of transition rules

CA model		Hybrid model	
Process	Number of transition rules	Alternate physical phenomenon	Number of transition rules
Application of transition rules for cell membrane formation	34	Gray Scott Model	8
Application of transition rules for division of information carriers	17	—	16
Application of transition rules for movement of information carriers	25	—	25
Application of transition rules for nuclear membrane formation	13	—	5
Total	89		54

## Conclusion

In this study, we constructed a model of a hybrid cellular automaton model. Our model displayed self-reproduction in a cell-like shape with few state transition rules. To reduce the number of transition rules, we considered not only the state transition rules but also the concentration diffusion in the Gray Scott model, in which the self-reproduction phenomenon emerges with certain parameters.

The future direction for this research includes the discovery of other sets of transition rules, identification of a way to derive transition rules automatically on the basis of a uniform law, and theoretical application of transition rules to particle collision.

Figure 8 shows the overall perspective of our artificial cell simulation. We believe that the transition rules of this model can be applied to the simulation of self-replication phenomena in a real dynamic chemical reaction environment. Initially, we plan to simulate cell division in a discrete particle reaction. It is relatively easy to replace state transition rules with collision/reaction rules of discrete particles. Next, we plan to simulate cell division in a continuous chemical reaction by converting discrete particles rules into chemical equations.

## References

### Book with Multiple Authors

- J. von Neumann (1966), *Theory of Self-replicating Automata*, University of Illinois Press  
 Meinhardt H, Prusinkiewicz P, Fowler D.(2003), *The Algorithmic Beauty of Sea Shells*. 3rd ed. New York: Springer.

### Journal Article

- Turing, A. M., (1952). "The chemical basis of morphogenesis", *Phil. Trans. Roy. Soc. Lond. B*, Vol.237 No.641, pp.37-72  
 Gray, P., and Scott, S. K., (1984). "Autocatalytic reactions in the isothermal, continuous stirred tank reactor: oscillations and instabilities in the system  $A+2B \rightarrow 3B$ ,  $B \rightarrow C$ ", *Chemical Engineering Science*, Vol.39, pp.1087-1097  
 Langton, C. G., (1989). *Artificial life: The proceedings of an interdisciplinary workshop on the synthesis and simulation of living systems*, pp.1-48, Addison-Wesley.  
 Pearson, J.E., (1993). "Complex patterns in a simple system", *Science*, Vol.216, pp.189-192  
 Reggia, J. A., Lohn, J. D., and Chou, H. H., (1998). "Self-Replicating Structures Evolution, Emergence, and Computation", *Artificial Life*, Vol. 4, No. 3, pp.283-302  
 Sipper, M. (1998). "Fifty years on Self-Replicating: An Overview", *Artificial Life*, Vol. 4, No. 3, pp.237-257  
 Mange, D., Stauffer, A., Peparado, L., and Tempesti, G., (2004). "A Macroscopic View of Self-Replication", *Proceedings of the IEEE*, vol.92, No.12, pp.1927-1928  
 Feugier, F. G., Mochizuki, A., Iwasa, Y., (2005). "Self-organizing formation of vascular system of plant leaves: co-orientation between auxin flux and pump proteins", *Journal of Theoretical Biology*, Vol.236, pp.266-375  
 Ono, N. and Ikegami, T.(2000). "Self-maintenance and Self-reproduction in an Abstract Cell Model", *Journal of Theoretical Biology*, Volume 206, Number 2, September 2000, pp. 243-253(11).  
 Tim J. Hutton: *The Organic Builder: A Public Experiment in Artificial Chemistry and Self-Replication*. *Artificial Life* 15(1): 21-28 (2009). 2007. 6  
 Tim J. Hutton, (2010). "Codd's self-replicating computer", *Journal Artificial Life archive*, Volume 16 Issue 2.  
 Ishida, T., (2010). "Simulating self-reproduction of cells in a two-dimensional cellular automaton". *Journal of Robotics and Mechatronics*, Vol.22, No.5, pp.669-676  
<http://www.fujipress.jp/finder/xslt.php?mode=present&inputfile=ROBOT002200050015.xml>

### Proceedings Paper

- Ishida, T., (2011). Simulation of self-reproduction phenomenon of cells in two-dimensional hybrid-cellular automata model, *The 16th International Symposium on Artificial Life and Robotics (AROB 16th '11)*, January 27-29, 2011, Beppu, Oita, JAPAN

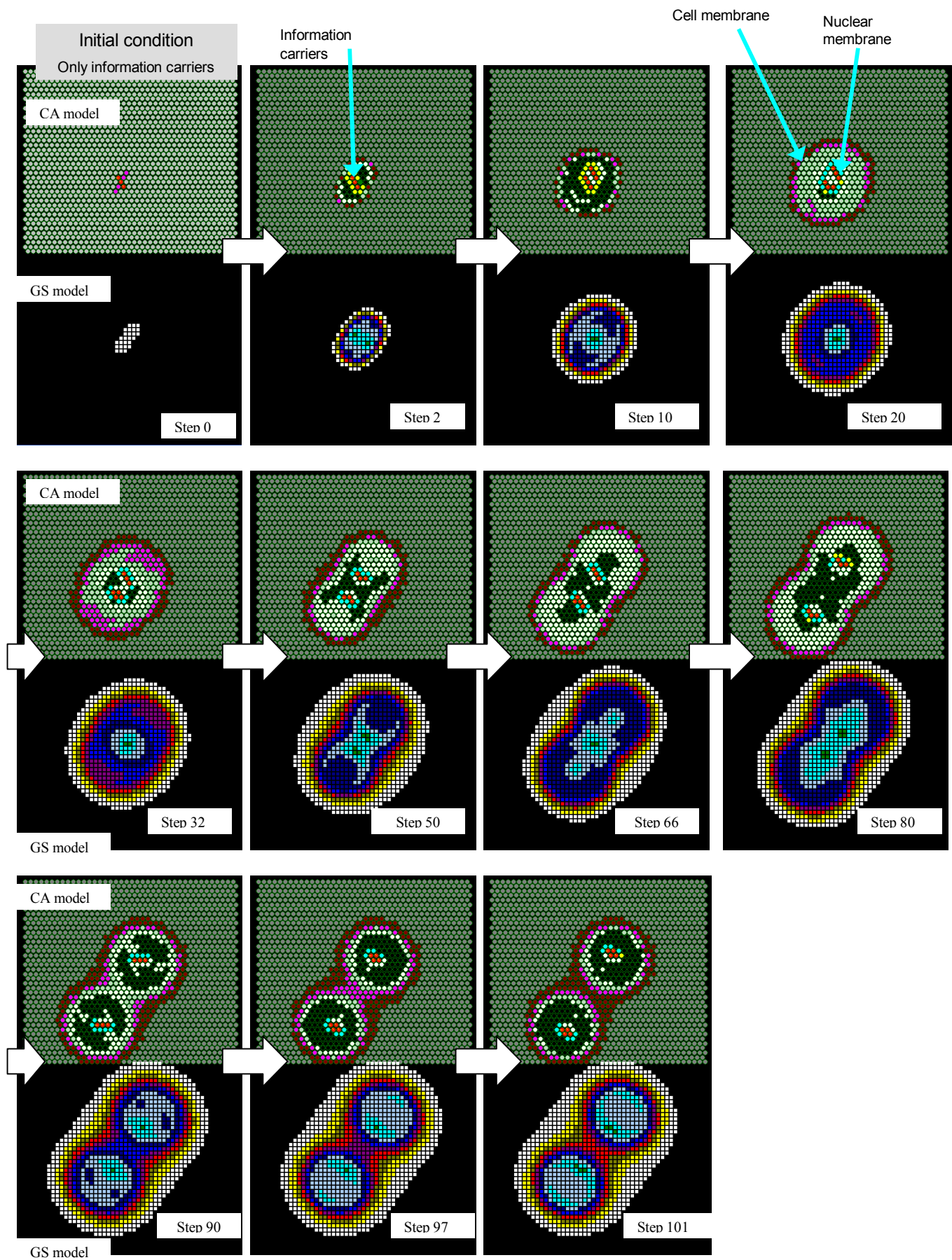


Figure 7 Results of hybrid cellular automaton model

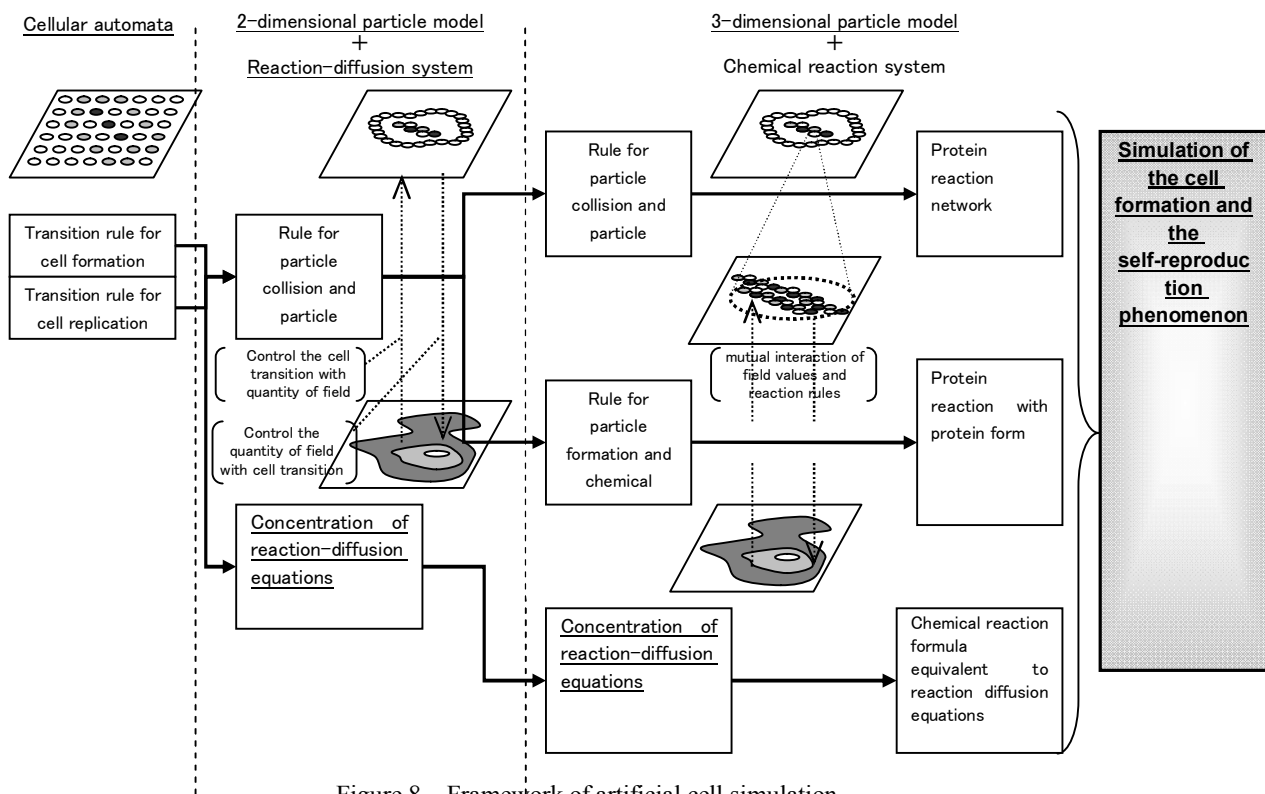


Figure 8 Framework of artificial cell simulation

# Social Learning and Evolution in a Structured Environment

Daniel Jones<sup>1</sup> and Tim Blackwell<sup>1</sup>

<sup>1</sup>Department of Computing, Goldsmiths, University of London, UK  
d.jones@gold.ac.uk, t.blackwell@gold.ac.uk

## Abstract

We survey the relationships between evolution, individual learning and social transmission within well-mixed and structured environments. With a novel individual-based simulation, we determine the regimes under which each mode of learning dominates, in terms of the environment's relative complexity and its rate of change. We show that social learning can give rise to a particularly potent form of the "Baldwin effect", wherein an organism develops an innate trait having first acquired it socially. We demonstrate that social learning is of increased significance in a structured environment.

## Introduction

To operate successfully in a Darwinian system, it is advantageous to possess maximal information about our environment. This is reflected in the functional information that all living creatures inherit via DNA, which codes for the set of functional characteristics most likely to benefit an organism in its future surroundings (Avery, 2003).

However, the environment which produced a parent is never quite the same as when its child is born. Ecological habitats are continually changing as their inhabitants consume and produce resources, with environments effectively co-evolving with their organisms. Inheritance is thus an intrinsically probabilistic process, which uses rules of thumb to provide the best possible solution given the expected habitat based on previous generations (Seth, 2007).

To optimally deal with uncertainty, all organisms exhibit some degree of *phenotypic plasticity* (West-Eberhard, 1989; Scheiner, 1993): the ability to alter behaviour or physiology in response to environmental conditions. By allowing some morphological decisions to be fixed later in an organism's lifetime, evolution can effectively defer decisions about functional specifics. This appears to be particularly prominent in fluctuating and heterogeneous environments, which are naturally less predictable (West-Eberhard, 1989).

## Evolution, learning and culture

The most radical form of phenotypic plasticity is behavioural learning, which can respond rapidly and flexibly

to novel stimuli based on prior experience. Learning can be considered as giving foresight to the blind process of evolution, by enabling an organism to search the fitness landscape around the point determined by its genotype (Belew, 1990; Borenstein et al., 2006). As Maynard Smith (1987) observes,

"...finding the optimal [solution] in the absence of learning is like searching for a needle in a haystack. With learning, it is like searching for the needle when someone tells you when you are getting close." (Maynard Smith, 1987, p762).

In this paper, we are concerned with two forms of learning: *individual exploration*, which we shall define as trial-and-error learning solely between an individual and its (abiotic) environment; and *social learning*, in which an organism acquires traits by observing or mimicking the behaviours of others (Lefebvre and Palameta, 1988). Countless species engage in social learning (Galef and Laland, 2005; Laland, 2004a), through mechanisms such as mimicry, teaching, and goal emulation. We shall here deal with a general case in which a trait is exhibited after observing another organism as a model (the "exemplar").

When evolutionary systems are extended with lifetime behavioural plasticity, we should expect some interesting interactions to arise. One which came to the attention of the first generation of evolutionary theorists after Darwin (Baldwin, 1896; Morgan, 1896) is the "Baldwin effect", a term coined half a century later (Simpson, 1953) after one of its progenitors, ironically in an attempt to discredit the theory.

The general pattern encapsulated within the Baldwin effect is as follows:

1. A population arises in which some trait  $P$  becomes beneficial.
2. Some individuals arise which, through their phenotypic plasticity, are able to learn  $P$ .
3. In some of these individuals, the trait  $P$  becomes innate (*genetic assimilation*).



With the assumption that innate behaviours are less costly than those which are plastic, we would then expect selective pressure to lessen on these particular learning capabilities: if we can do it by nature, we no longer need to be able to learn it (West-Eberhard, 2003).

The status and prevalence of genetic assimilation within real-world ecosystems is as yet unresolved, and subject to some controversy (Pigliucci et al., 2006). Due to its onerous requirements – a species sufficiently advanced to partake in social learning, bred over a sufficient number of generations for a trait to become genetically incorporated – it is difficult to observe via *in vivo* studies, though Waddington’s (1953) “veinless” study elegantly demonstrates its biological plausibility. It is, therefore, a well-suited candidate for *in silico* experiments.

### Theoretical studies of social learning

A large body of theoretical work has been developed at the confluence between evolution, learning and cultural transmission (Cavalli-Sforza and Feldman, 1981; Boyd and Richerson, 1985; Wakano et al., 2004). The watershed work was a computational model by Hinton and Nowlan (1987, henceforth ‘H&N’), who extended binary genetic algorithms with an undefined third value, whose outcome is determined by lifetime learning. Though intentionally simplistic, this model effectively demonstrated the “needle in a haystack” function of learning as a dowsing rod to guide evolution towards discontinuous fitness peaks.

Belew (1990) and Best (1999) have extended H&N with differing forms of cultural transmission, both in a well-mixed environment, incorporating oblique and horizontal forms of social exchange. While Belew models cultural exchange as a bias towards higher fitness, we will follow Best as treating it as a more neutral form of behavioural mimicry, in which an organism may imitate deleterious as well as adaptive behaviours.

Models of social transmission within a spatial environment include work by Boyd and Richerson (1988), Lowen (1996) and Borenstein (2003). A consensus view has emerged that sociality is of benefit within structured environments. We wish to extend these analyses to survey the regimes under which each mode of learning excels, and whether unforeseen mixed strategies may come to the fore given a heterogeneous, individual-based model.

We also wish to model the scenario conjectured by Papineau (2005), who posits that the Baldwin effect may become significantly more prominent when bolstered with social learning. This can be roughly encapsulated by the inequality:

$$p(G) \ll p(L) \ll p(S) \quad (1)$$

Where  $p(G)$  is the probability of exhibiting a trait innately,  $p(L)$  is the probability of learning it through ex-

ploration, and  $p(S)$  is the probability of acquiring the trait through social learning. Quite simply, wherein it is effectively impossible to acquire a functional trait  $P$  through evolution – perhaps because it is comprised of multiple sub-traits, which are jointly necessary to reap a fitness benefit – this process may be somewhat more likely when lifetime learning is possible, and even moreso when social learning enables organisms to share traits.

This argument, though intuitively sound, is thus far based on heuristic assumptions. The following model is intended to quantitatively explore situations in which a social Baldwin effect can take place, and particularly those in which combination strategies can arise: evolved individuals can exhibit both individual and social learning in proportion. We are furthermore interested in how these phenomena interact in a context which is explicitly spatial, a combination which has not yet received significant attention.

### Model specification

We will now describe the components of the individual-based model used to explore these ideas<sup>1</sup>. An *environment*  $E$  consists of a  $B$ -bit string, representing a ‘target’ task:  $E \in \{0, 1\}^B$ . The current environmental state can therefore be considered as a vertex on an  $B$ -dimensional hypercube.

It is inhabited by a population of  $N$  agents, each of which has the following properties:

- $b_{evo}, b_{exp}, b_{soc} \in [0, 1]$  – *behavioural traits* determining the propensity towards evolutionary instinct, individual exploration, and social learning. These are collectively normalised to sum to unity.
- $g \in \{0, 1\}^B$  – *genotype*, a  $B$ -bit string corresponding to the capability to fulfil the environment’s target task.
- $p \in \{0, 1\}^B$  – *phenotype*, a  $B$ -bit string initially equal to  $g$ , but subject to modification through individual and social learning. If  $p$  is equal to  $E$  then the agent’s fitness is maximised.
- $\Omega$  – current metabolic state, initialised to a constant  $\Omega_0$ .

An agent’s current phenotype determines how well it complies with the environment’s demands, based on its Hamming distance from  $E$ . Its metabolic state determines the extent to which it has ‘grown’ throughout its lifetime.

### Actions and learning

Every timestep, each agent selects a behavioural mode according to a weighted random of  $\{b_{evo}, b_{exp}, b_{soc}\}$ :

- $b_{evo}$  – act according to the agent’s current phenotype
- $b_{exp}$  – act according to the agent’s current phenotype, with  $\beta$  bits toggled at random

<sup>1</sup>For all subsequent parameter values, see the *Methods* section.



- $b_{soc}$  – act according to the agent’s current phenotype, with  $\beta$  bits copied from a neighbour using roulette wheel selection weighted by  $\Omega$ . With a probability  $p_{noise}$ , each of these bits may be copied erroneously (that is, toggled from  $0 \rightarrow 1$  or  $1 \rightarrow 0$ ). This models the inaccuracy present in real-world imitative learning: a behaviour may be only partially observed, or reproduced incorrectly.

If  $b_{exp}$  or  $b_{soc}$  is employed *and* the resultant action gives a higher payoff than the agent’s own current phenotype, the corresponding bits in  $p$  are replaced by the new action: discovering (or imitating) a successful new trait results in its being incorporated into the agent’s roster. This reflects phenotypic plasticity, where  $\beta$  is the limiting factor on the rate at which new skills can be acquired.

In the case of  $b_{soc}$ , weighting the exemplar by their  $\Omega$  value reflects a tendency towards mimicking those organisms which are perceived as being fittest. This is described by Laland (2004b) as a “*copy-successful-individuals*” strategy, as observed in avian, chimpanzee and bat societies.

The agent’s metabolism is then modified according to the following update rule:

$$\Delta\Omega = \left(1 - \frac{H(p, E)}{B}\right)^{\alpha-1} \quad (2)$$

where  $H$  denotes the Hamming distance between two bit strings. The exponential of  $\alpha$  is used to determine the fitness differential between perfect and almost-perfect task fulfilment: a lower value of  $\alpha$  means that payoffs fall more rapidly with distance. With  $\alpha = 1$ , scaling is linear in distance.

In general, if an agent’s  $g$  matches precisely the tasks specified in  $E_i$ , its metabolism will increase by the maximal value of 1. If  $g$  is precisely the complement of  $E_i$ , its metabolism will increase by 0.

Taken as a population mean, the metabolic rate  $\Delta\Omega$  can be considered as a measure of *fitness*, as it is directly proportional to reproductive rate. We will subsequently use the terms interchangeably.

## Reproduction

When an agent’s metabolism  $\Omega$  reaches the value  $2\Omega_0$ , the agent reproduces asexually. Its offspring has an identical genotype, subject to each bit of  $g$  mutating with small probability  $p_{mut}$ . Behavioural traits  $b_{evo}$ ,  $b_{exp}$ ,  $b_{soc}$  are modified by a zero-mean Gaussian noise function, standard deviation  $\mu$ , and clipped to  $[0, 1]$ . These are again collectively normalised to unity. The child replaces a member of the population selected uniformly randomly, and its parent’s  $\Omega$  value is reset to  $\Omega_0$ .

Sexual recombination was considered as a reproductive strategy. Kauffman (1993) observes that recombination is an effective method of finding ‘middle ground’ locations between points on a complex fitness landscape. However,

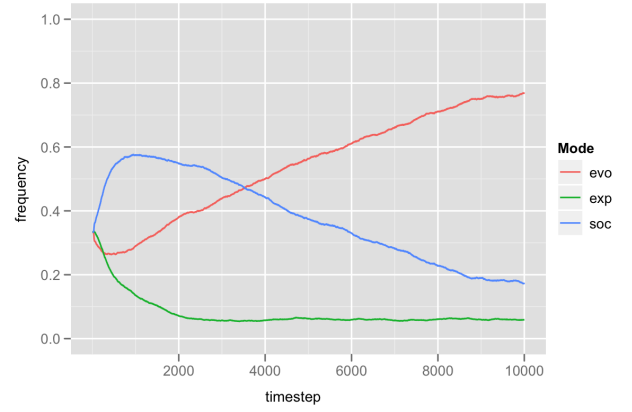


Figure 1: Distribution of behaviours in a static environment, averaged over 25 simulation outcomes.

given our single-peaked landscape, we focus on clonal reproduction for the sake of simplicity. A number of recombinative trials indicated that the results would not be qualitatively different.

## Results

The results of this model are presented in incremental form, with processes introduced gradually. The motivation behind this approach is to understand pairwise interactions between adaptive mechanisms. By doing so, we hope to fully understand the causal basis behind the emergent phenomena.

### Static environment

We initialise the environment’s task to  $1^B$  for clarity (following Hinton and Nowlan (1987)). Behavioural traits are initialised to uniformly random values, and the population left to evolve.

The changing distribution of behavioural traits over time is shown in Figure 1, as averaged over multiple iterations (see Methods). At step 0, the frequency of each is  $\frac{1}{3}$ , indicating the initial uniformly random distribution of behavioural modes.

The dynamics can subsequently be divided into three phases. (i) Between steps 1–4000, the population is dominated by social learners, with a generally low level of genotypic fitness meaning that a costlier but fitter social learning is preferable. (ii) From steps 4000–9000, the trait has been assimilated into the genotype of the majority, and so innate  $b_{evo}$  agents outcompete their costlier plastic rivals. (iii) Beyond step 9000, a stable optimum is reached.

Sharpening these costs by reducing payoff scaling factor  $\alpha_0$  results in a more rapid convergence to a predominantly  $b_{evo}$  population. This also reduces the effectiveness of lifetime learning, of course, which introduces a penalty in fluctuating environments.

This is a clear example of the Baldwin effect. Phenotypically plastic individuals first outcompete their peers (*i*) as they scramble to higher fitness through learning and social exchange, and are subsequently replaced (*ii*, *iii*) by innate mutants, who do not bear the costs of exploration.

A repeated trial wherein all agents begin with a genotype of  $1^B$  reveals, as anticipated, that they continue to maintain a stable state with only low levels of social and individual learning.

### Static environment with restricted strategies

The above experiment was repeated with a fixed trait mutation factor of  $\mu = 0$  and initial behavioural traits restricted to specific combinations: either pure evolutionary learning ( $b_{evo}$ ), or evolution plus learning ( $b_{evo} + b_{exp}$ ), or evolution plus social learning ( $b_{evo} + b_{soc}$ ), or all three traits in combination.

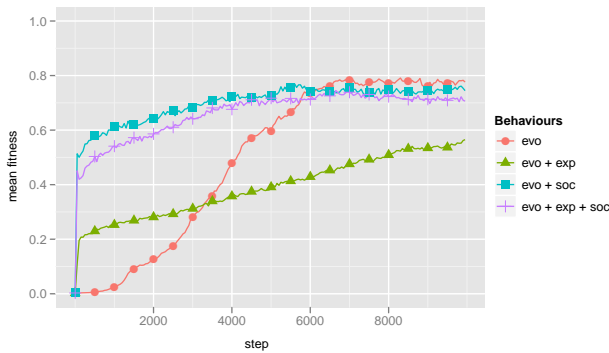


Figure 2: Convergence rates with four different strategies:  $b_{evo}$ ,  $b_{evo} + b_{exp}$ ,  $b_{evo} + b_{soc}$ , and  $b_{evo} + b_{exp} + b_{soc}$ .

Figure 2 depicts the relative effectiveness of each strategy in a static environment, plotting the global mean fitness (that is,  $\Delta\Omega$ ) over a number of generations. The key indicators of success are the convergence rate and the value to which the population converges.

All four strategies eventually converge around the same peak of 0.8. The times taken to do so, however, are markedly different. Notably, evolution plus learning takes substantially more time to converge than pure evolution alone, and continues to trail throughout the simulation. This confirms the findings of Borenstein et al (2008) that, in a static, unimodal fitness landscape, individual learning actually serves to slow convergence rates.

With social learning, convergence times are markedly more rapid, reaching a mean fitness of 0.5 in less than half the time as evolution or evolution plus learning.

### Static environment with single perturbation

Here, the scenario was repeated as per the Static Environment case, with with an environmental perturbation induced at step 10000: each of its bits were flipped according to a

probability  $p = 0.5$ . As indicated in Figure 3, this change results in a temporary increase in social and exploratory learners, bringing up phenotypic fitness through plasticity whilst evolution takes time to work out the necessary series of mutations.

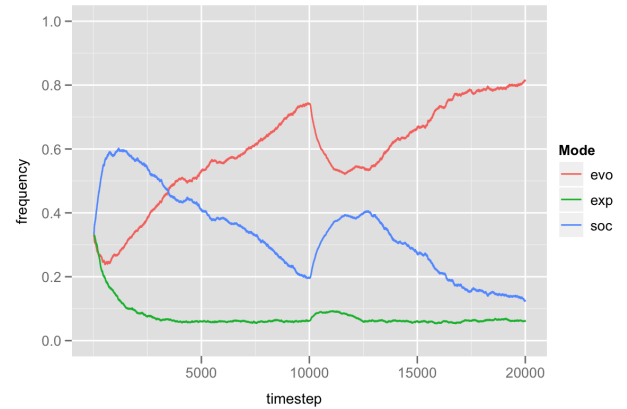


Figure 3: A single perturbation occurs at  $t = 10000$ . Subsequently, agents are selected for increased social and exploratory learning tendencies.

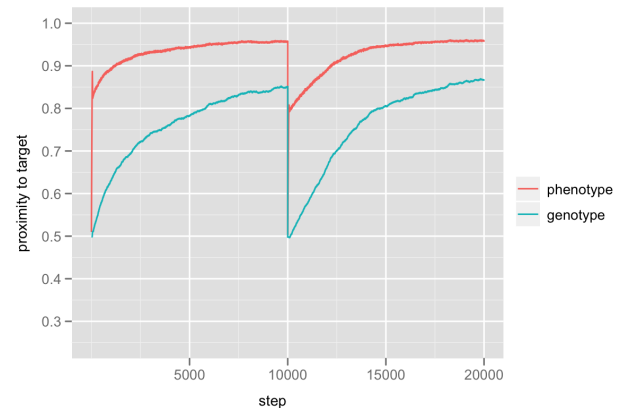


Figure 4: Genotypic and phenotypic fitness after perturbations.

This further demonstrates Baldwin-like phenomena, and moreover with a *social* focus: whilst a small proportion of individuals respond to environmental change by switching to individual exploration, the predominant trend is to rely on social learning, observing the behaviour of others to maximise fitness.

### Fluctuating environment

We now extend the above by introducing irregular environmental fluctuations. Each time step, a single bit of the environmental task may be toggled, according to a small probability  $p_{switch}$ . A value of  $p_{switch} = 0.01$  reflects an ex-

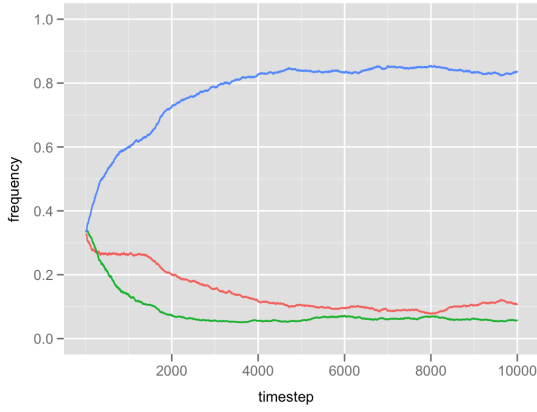


Figure 5: With a regularly fluctuating environment ( $p_{switch} = 0.01$ ), a social learning strategy is more frequently adopted.

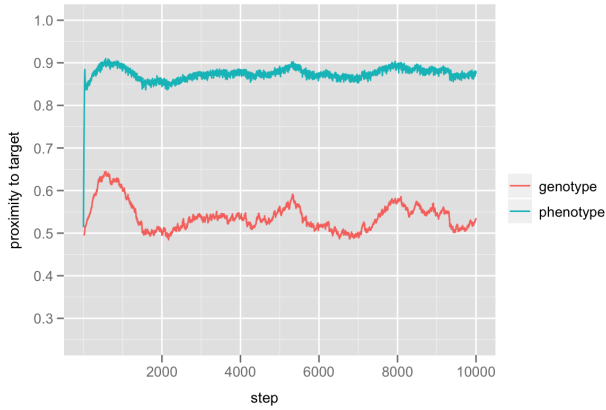


Figure 6: With a higher reliance on phenotypic plasticity, genetic selection pressure is lower, and so genotypic constitution drifts.

pected period of 100 timesteps between fluctuations. With an initial metabolism  $\Omega_0 = 10$  and a typical  $\Delta\Omega = 0.5$ , the environment could be expected to fluctuate once every 5 agent-lifespans.

The optimal combination of strategies is markedly different than in a fixed environment (Figures 5 and 6). Social learning dominates, reflecting the benefit of a faster adaptive rate with changing fitness targets.

Convergence patterns are also markedly different (Figure 7). In a rapidly changing environment ( $p_{switch} = 0.005$ ), no strategy attains a mean fitness of above 0.7: even with the ability to mimic successful peers, it is difficult to maintain a high performance level in the face of continuous change. Social learning is frontrunner once more, with  $b_{evo} + b_{exp}$  significantly outperforming pure evolution. This reflects the advantage in random trials when an organism's genome is lagging behind the rate of change of its environ-

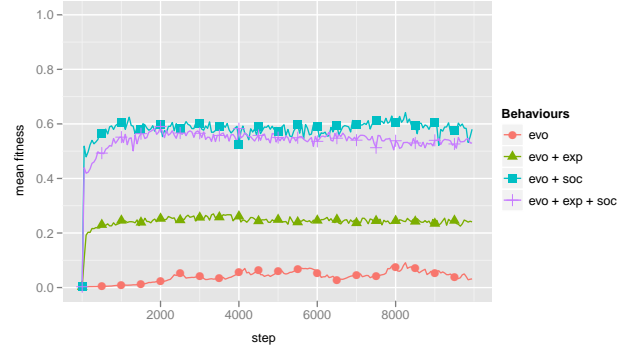


Figure 7: Convergence rates are markedly different within a fluctuating environment ( $p_{switch} = 0.005$ ).

ment.

### Environmental rate and complexity

To gain fuller insight into the relative strengths of individual, social and exploratory learning in fluctuating environments, we carried out an array of simulations over a range of rates of change ( $p_{switch} \in [5 \times 10^{-6}, 0.5]$ ) and environmental complexities ( $B \in [1, 2048]$ ). Each permutation of  $p_{switch}$  and  $B$  was executed for  $10^5$  timesteps, and a snapshot taken of the final distribution of behavioural traits. These are mapped in Figure 8, with the dominance of each trait demonstrated by its share of the pie chart at the given (*complexity, rate*) combination.

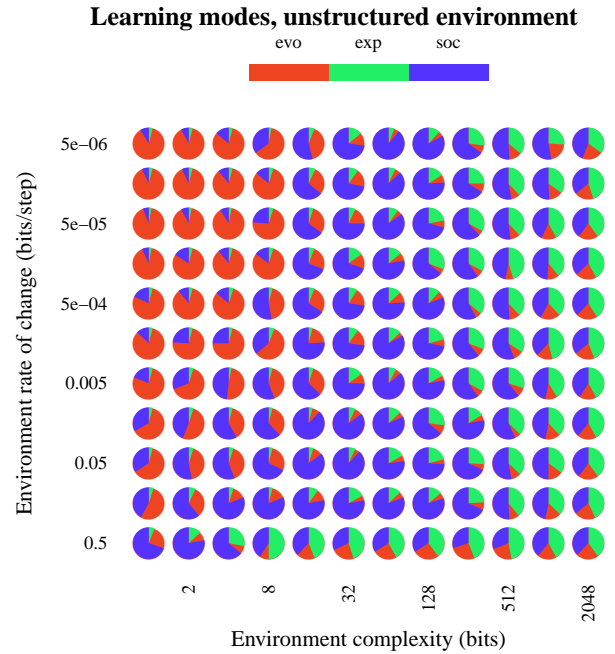


Figure 8: Dominant learning modes at equilibrium, varying dimensionality and rate of change of environment.

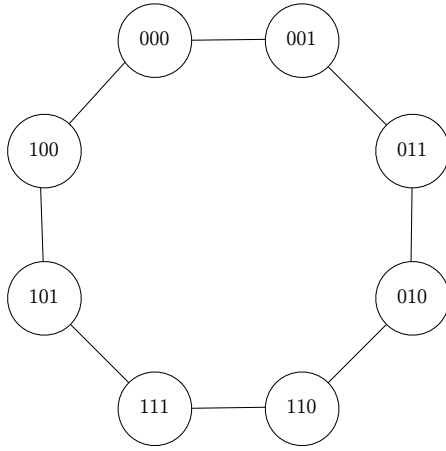


Figure 9: Ring of environmental ‘cells’. Tasks are numbered according to Gray code: note that adjacent cells have 1 bit difference.

At low rates of change and in simple environments, the population demonstrates a significantly greater mean growth rate, with a clear prevalence of  $b_{evo}$ . As *either* rate or complexity increase, strategies become more mixed, with a trend towards social learning at median values of each. A greater amount of noise in the results suggests that selection pressures per are weaker, leading to more vulnerability to stochastic variation.

At very high rates of change or complexity, a sudden increase of  $b_{exp}$  dominance is evident. This is relatively simple to interpret in the former case: if the environment is changing faster than information can percolate through a social group, then even social learning is inferior to individual trial and error.

The benefit of learning in very complex environments is less clear; even in a virtually static environment ( $p_{switch} = 5 \times 10^{-6}$ ), exploration exceeds innate strategies for  $B = 2048$ . Analysis reveals that the fitness ( $\Delta\Omega$ ) in these regimes is uniformly low: given the rapid fitness falloff due to  $\alpha$ , neither evolution nor social learning are fit to find suitable values. With such a large parameter space, the optimal resort is simply bit-wise trial and error.

### Structured environment

We now extend the model by introducing a form of spatial structure. The single environment is replaced by a 1-dimensional ring of  $L$  environmental ‘cells’, each with a distinct population and set of tasks (Figure 9). Inhabitants of each cell can only interact with each other. As before, the size of the total metapopulation remains constant at  $N$ .

Each environmental cell has a single neighbour on each side, with the rightmost cell wrapping around to the leftmost. To introduce correlation between the task structure of neighbouring cells, integer sequences were produced us-

ing Gray code, a base-2 numeric encoding in which any two adjacent integers have a Hamming distance of 1. A further property of Gray code sequences is that they are cyclical, with the first and final integer of any  $2^N$ -length sequence also one bit apart. It is possible, therefore, to produce integer rings with pairwise Hamming distance of 1.

During a timestep, a agents may move from their current location to a neighbouring cell with a small probability  $p_{move}$ . Evolutionary and individual learning are unaffected; social learning, however, is now restricted to exemplars within the agent’s current location.

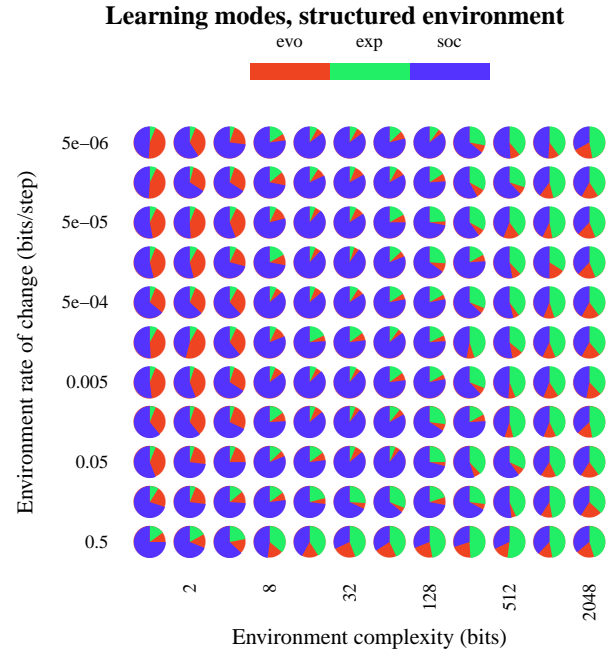


Figure 10: Dominant learning behaviours in a structured environment with migration.

From Figure 10, we can see that the overall distribution of learning patterns is similar: in static, simple environments, innate behaviour is commonplace, moving towards social learning in more complex and fluctuating contexts.

With rapid fluctuations, exploratory learning still excels, but it appears to have slightly less prevalence in environments with a large  $B$  value. This appears to be due to what we will call the “*local specialist*” effect: in a well-mixed, complex environment, there are a large range of behaviours to mimic, drawn from a large variety of sources. Even if we select our exemplar wisely, we may still mimic the wrong behaviour, as they too will be employing random search to test out new tasks.

In a structured environment, conversely, we have smaller number of local neighbours to mimic. With the roulette-wheel mechanism used to select exemplars, a smaller population also means a higher likelihood of selecting an highly-

ranked target. Combined with the fact that selection pressure still operates by removing the weakest agents of the *global* population, this means that positive behaviours are disseminated and adopted rapidly within individual cells, giving rise to social ‘specialist’ cliques.

### Structured environment with migration

In this scenario, we remove environmental fluctuations, and instead vary  $p_{move}$ : the rate at which migration occurs. Figure 11 depicts this new distribution, with its Y-axis representing the rate of migration, over the same range as the fluctuation rate was previously plotted. The only variation that an agent will experience in its environment is when it moves from cell to neighbouring cell, so this can effectively be considered analogous to our previous environmental fluctuations.

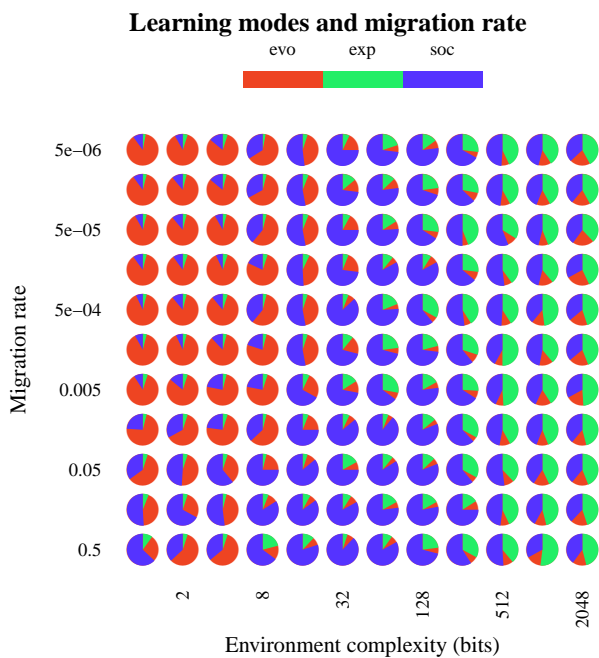


Figure 11: Dominant learning behaviours by migration rate and environmental complexity.

Learning strategies do not appear to correlate significantly with movement rates, despite the fact that movement between cells *does* effectively change the environment that an agent experiences. However, a significant difference takes place at high movement rates. Rather than resorting to individual trial and error, agents make greater use of social learning. This may be interpreted as a more focused version of the local specialist effect; in a static environment with frequent migration, we would expect the rapid dissemination of local knowledge to become of paramount importance.

In other words, if an agent is commonly moving from environment to environment, the most effective way to obtain

information about novel functions is to mimic the locals. This has both logical and biological plausibility.

### Discussion

We have seen that three discrete regimes appear within varying classes of environment, each favouring different forms of learning. Within static environments, innate behaviour excels; within rapidly-changing environment, exploratory behaviour comes to the fore. Social behaviour, conversely, fills the gap between the two.

Beyond this, social transmission serves to inform and drive subsequent evolutionary behaviour, with what Papineau (2005) terms a “social Baldwin effect”. Our results suggest that this may play a pivotal role in the aftermath of major environmental changes – which, in ecosystems wherein organisms act as background to other organisms, may also correspond to the aftermath of major *ecological* changes.

In a structured environment, we have seen that successful behaviours are disseminated rapidly, due to reliance on smaller, focused groups of ‘specialists’ in each location. With greater environmental complexity, these local effects are amplified yet further.

### Methods

Simulation results are averaged over 25 iterations to minimise stochastic fluctuations. Default variable values are given below.

Variable	Value	Comments
$N$	256	Population size
$L$	32	Number of spatial locations
$B$	32	Number of bits per task
$\Omega_0$	10	Initial metabolic state
$\alpha$	0.1	Rate of fitness dropoff based on task proximity
$\beta$	1	Maximum number of bits learned per timestep
$\mu$	0.01	s.d. of mutation as applied to $b_{evo}, b_{exp}, b_{soc}$
$p_{switch}$	0.01	Probability of a single environmental fluctuation
$p_{noise}$	0.25	Probability of incorrect observation during mimicking
$p_{mut}$	0.01	Probability of sustaining a mutation per gene
$p_{move}$	0.1	Probability of migrating to a neighbouring cell

### Acknowledgements

The authors would like to thank Christophe Rhodes, Tim Taylor and Mark d’Inverno for their insightful comments.

This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/P503418/1.

## References

- Avery, J. (2003). *Information Theory and Evolution*. World Scientific.
- Baldwin, J. M. (1896). A new factor in evolution. *The American Naturalist*, 30(354):441–451.
- Belew, R. K. (1990). Evolution, learning and culture: Computational metaphors for adaptive algorithms. *Complex Systems*, 4:11–49.
- Best, M. L. (1999). How culture can guide evolution: An inquiry into Gene/Meme enhancement and opposition. *Adaptive Behavior*, 7(3-4):289–306.
- Borenstein, E., Feldman, M. W., and Aoki, K. (2008). Evolution of learning in fluctuating environments: when selection favors both social and exploratory individual learning. *Evolution; international journal of organic evolution*, 62(3):586–602.
- Borenstein, E., Kendal, J., and Feldman, M. (2006). Cultural niche construction in a metapopulation. *Theoretical Population Biology*, 70(1):92–104.
- Borenstein, E. and Ruppín, E. (2003). Enhancing autonomous agents' evolution with learning by imitation. *Journal of Artificial Intelligence and the Simulation of Behavior*, 1(4):335–347.
- Boyd, R. and Richerson, P. J. (1985). *Culture and the Evolutionary Process*. The University of Chicago Press.
- Boyd, R. and Richerson, P. J. (1988). *An Evolutionary Model of Social Learning: The Effects of Spatial and Temporal Variation*. Erlbaum, Hillsdale, NJ.
- Cavalli-Sforza, L. L. and Feldman, M. W. (1981). *Cultural Transmission and Evolution: A Quantitative Approach*. Princeton University Press.
- Galef, B. G. and Laland, K. N. (2005). Social learning in animals: Empirical studies and theoretical models. *BioScience*, 55(6):489–499.
- Hinton, G. E. and Nowlan, S. J. (1987). How learning can guide evolution. *Complex Systems*, 1:495–502.
- Kauffman, S. A. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, USA, 1 edition.
- Laland, K. N. (2004a). Extending the extended phenotype. *Biology and Philosophy*, 19(3):313–325.
- Laland, K. N. (2004b). Social learning strategies. *Learning & Behavior*, 32(1):4–14.
- Lefebvre, L. and Palameta, B. (1988). Mechanisms, ecology, and population diffusion of socially-learned, food-finding behavior in feral pigeons. In Zentall, T. R. and Galef, B. G., editors, *Social Learning: Psychological and Biological Perspectives*, pages 141–164. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Lowen, C. B. and Dunbar, R. I. M. (1996). Cultural evolution in a structured population. In *ESRC Centre on Economics Learning and Social Evolution*.
- Maynard Smith, J. (1987). When learning guides evolution. *Nature*, 329:761–762.
- Morgan, C. L. (1896). On modification and variation. *Science*, 4:733–740.
- Papineau, D. (2005). Social learning and the baldwin effect.
- Pigliucci, M., Murren, C. J., and Schlichting, C. D. (2006). Phenotypic plasticity and evolution by genetic assimilation. *The Journal of experimental biology*, 209(Pt 12):2362–2367.
- Scheiner, S. M. (1993). Genetics and evolution of phenotypic plasticity. *Annual Review of Ecology and Systematics*, 24:35–68.
- Seth, A. K. (2007). The ecology of action selection: insights from artificial life. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 362(1485):1545–1558.
- Simpson, G. G. (1953). The baldwin effect. *Evolution*, 7(2).
- Waddington, C. H. (1953). Genetic assimilation of an acquired character. *Evolution*, 7(2):118–126.
- Wakano, J., Aoki, K., and Feldman, M. (2004). Evolution of social learning: a mathematical analysis. *Theoretical Population Biology*, 66(3):249–258.
- West-Eberhard, M. J. (1989). Phenotypic plasticity and the origins of diversity. *Annual Review of Ecology and Systematics*, 20.
- West-Eberhard, M. J. (2003). *Developmental Plasticity and Evolution*. Oxford University Press, USA, 1 edition.



# Evolutionary Dynamics of GAs in a Simple Model with Dynamical Environment and Neutrality

Yoshiaki Katada<sup>1</sup>

<sup>1</sup>Setsunan University, Osaka 572-8508, JAPAN  
katada @ ele.setsunan.ac.jp

## Abstract

Biological organisms have various mechanisms of coping with the dynamical environments in which they live. Recent papers in computational biology show that individuals reside in deferent regions of neutral networks according to environmental variation. This work investigated evolutionary dynamics of GAs in dynamical environments with neutrality using a simple model. The evolutionary dynamics observed were consistent with those observed in the experiments of biological evolution, confirming that the genotype distributions change depending on the rates of environmental variation as well as mutation.

## Introduction

The Neutral theory (Kimura, 1983) was developed by Motoo Kimura in the 1960s. Neutrality is caused by highly redundant mappings from genotype to phenotype or from phenotype to fitness. Based on this, it was reported that biological organisms make good use of genetic mechanisms which do not appear in phenotype to adapt to environmental variations on the evolutionary time scale.

The effects of neutrality has been discussed so much in the EC community especially since Harvey introduced the concept of neutral networks (Harvey and Thompson, 1996). These researches can be classified into two types as follows. The former researches are based on redundant mappings from phenotype to fitness, where neutral networks are included in a problem itself. Examples would be the evolution of neural network controllers in robotics (Harvey, 1997; Smith et al., 2001) and on-chip electronic circuit evolution (Thompson, 1996; Vassilev et al., 2000). In these researches, evolutionary dynamics are investigated (Barnett, 1997; Newman and Engelhardt, 1998; van Nimwegen et al., 1999; Katada et al., 2004) or the degree of neutrality in fitness landscapes is estimated (Smith et al., 2002; Katada and Ohkura, 2006). The latter based on redundant mappings from genotype to phenotype, where redundancy, that is, neutrality has been intentionally incorporated by EC researchers for problems where redundancy is largely absent to improve the performance of artificial evolution (Ohkura and Ueda,

1999; Ebner et al., 2001; Knowles and Watson, 2002; Rothlauf and Goldberg, 2003).

To the best of my knowledge, in the former type of research, neither evolutionary dynamics nor useful genetic operators in *dynamical environment* has been investigated. Independently of neutrality, representations of polyploid model in dynamical environment have been investigated where useful genes in previous environments are preserved in some kind of memories (Branke, 2001). Apparently, the feature of polyploidy is a redundant genetic material, that is, redundant mappings from genotype to phenotype. However, it seems likely that there is no research that investigate this from the view point of neutrality.

GP, whose evolved programs include many introns and functionally redundant parts, would be classified into the former research. That is why some GP researchers have claimed the importance of neutrality in recent years (Yu and Miller, 2006; Miller, 2009; Vanneschi, 2009).

Recent papers in computational biology show that individuals reside in deferent regions of neutral networks according to environmental variation. Meyers et al. (2005) analyzed evolution in a periodically changing environment using a simple model and a codon model where a locus has several alleles and some of them are functionally equal, and reported as follows: When environmental variations are rare, most individuals are located in the center of the neutral network with the highest fitness value in each environment preparing for detrimental mutation (Fig. 1(a)). This phenomenon is called *genetic robustness*. When the rates of environmental variation are intermediate, most individuals are located in the edge of the neutral network in order to obtain a new phenotype which can adapt to an alternating environment with a few mutations (Fig. 1(b)). This is called *genetic potential*. When the rates of environmental variation are high, they are settled in a phenotype with an intermediate fitness value in both environments (Fig. 1(c)). This would mean that they have tolerances and adaptivity for both environments but would never go to extremes. This is called *organismal flexibility*.

Based on these knowledges, Yu (2007) investigated evolu-



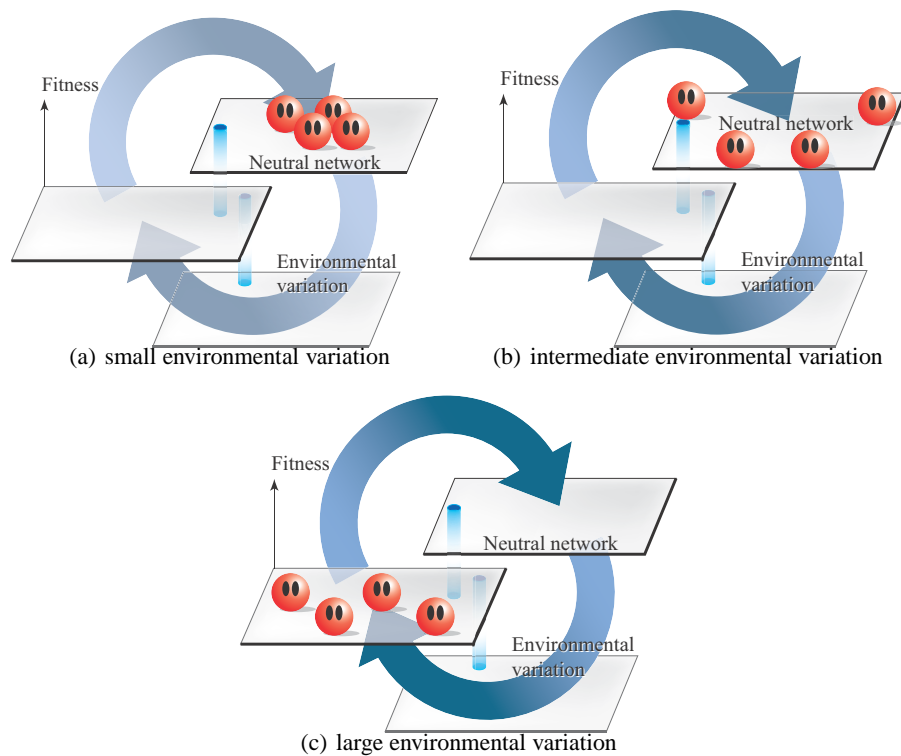


Figure 1: Distribution of individuals due to environmental variation

tionary dynamics of the GP in a boolean parity problem under environmental variations. It was reported that when the variation rate is high, the length of a program tree became long, that is, the effective mutation rate per individual became high, and when the variation rate is low, the length became short, that is, the effective mutation rate per individual became low. Yu (2007) claimed that when the variation rate becomes high, individuals of the GP tend to be located in the edge of the neutral network because the effective mutation rate per individual becomes high and individuals easily change their phenotype. However, we have trouble defining a neutral network on GPs due to its representation. Therefore, it is difficult to discuss directly the consistency of the obtained results in the GP to the computational biology because we need the concept of location on a neutral network for them.

Based on these results, the question arises as to whether we can get the same kind of dynamics of GAs in dynamical environments with neutrality because neutral networks have been found in GAs with highly redundant mappings from phenotype to fitness. In the case of GAs with redundant mappings from genotype to phenotype (including polyploidy), we would get the same kind of results on the “GPs” mentioned above because it would be difficult for the GAs to devise a neutral network<sup>1</sup> and effective mutation rates of

<sup>1</sup>It is possible to define a neutral network in GAs with redundant

them are variable.

This paper focuses on the former case, where GAs with redundant mappings from phenotype to fitness (more precisely, genotype to fitness) that can form neutral networks and investigates evolutionary dynamics of them in a simple model by varying the rates of environmental variation and the mutation rate. The paper is organized as follows. The next section describes a neutral network in a mathematical form. Section III describes a simple model of dynamical environments with neutrality where evolutionary dynamics of GAs is investigated. Section IV gives the results of our computer simulations. Section V discusses the consistencies with the results obtained in computational biology. Conclusions are given in the last section.

### A Formal Definition of a Neutral Network

Katada and Ohkura (2009) defined a neutral network in a mathematical formula. The details are as follows;

In this study, it is assumed that genotypes are represented as binary strings and the length of them is fixed. Thus, the genetic distance between two different genotypes ( $x^g, y^g \in \Phi_g, x^g \neq y^g, \Phi_g$ : the set of genotypes determined by the length of the genotype,  $l$ ) is described by the Hamming dis-

mappings from genotype to phenotype (See the next section) but difficult to make neutral networks emerge from genotype space in which neutrality is intentionally incorporated as mentioned earlier.

tance between them,  $H(x^g, y^g)$ . Thus,  $\min H(x^g, y^g)$  is the smallest unit of mutation. For binary representations,  $\min H(x^g, y^g) = 1$ .

Based on the above consideration, I describe a neutral network caused by redundant mappings from genotype to phenotype in a mathematical form. At first, two individuals,  $x^g$  and  $z^g$ , are connected,  $x^g \sim z^g$ , if there exists  $\{x_i^g\}_{i=0}^n \subset \Phi_g$ , s.t.

1.  $x^g = x_0^g, z^g = x_n^g$ ,
2.  $f_g(x_i^g) = f_g(x_{i+1}^g)$ ,
3.  $H(x_i^g, x_{i+1}^g) = 1$ ,

where  $f_g$  is the mapping from genotype to phenotype,  $f_g : \Phi_g \rightarrow \Phi_p$ , and assumed to be surjective and not injective.  $\Phi_p$  is defined as the set of phenotypes.

Thus, a neutral network of a genotype  $z^g$  is

$$\Phi_g'(z^g) = \{x^g \in \Phi_g | x^g \sim z^g\}. \quad (1)$$

We can extend this definition to redundant mappings from phenotype to fitness.

Two individuals,  $x^g$  and  $z^g$ , are connected,  $x^g \sim z^g$ , if there exists  $\{x_i^g\}_{i=0}^n \subset \Phi_g$ , s.t.

1.  $x^g = x_0^g, z^g = x_n^g$ ,
2.  $(f_p \circ f_g)(x_i^g) = (f_p \circ f_g)(x_{i+1}^g)$ ,
3.  $H(x_i^g, x_{i+1}^g) = 1$ ,

where  $f_p$  is the mapping from phenotype to fitness,  $f_p : \Phi_p \rightarrow \Phi_f$ , and assumed to be surjective and not injective.  $\Phi_f$  is defined as the set of fitness values. Addition to this assumption, there would be two cases on  $f_g$ , which is either bijective, or surjective and not injective. In both cases, however,  $f_p \circ f_g$  is surjective and not injective only if  $f_p$  is surjective and not injective. Thus, a neutral network of a genotype  $z^g$  is described in the both cases as follows:

$$\Phi_g^*(z^g) = \{x^g \in \Phi_g | x^g \sim z^g\}. \quad (2)$$

These may seem to be cumbersome at first. But this elegant definition allows us to understand clearly a setting for computational experiments in the following sections.

### Simple Model with Dynamical Environment and Neutrality

In this study, computer simulations were conducted in order to compare evolutionary dynamics of GAs with those observed in the experiments of biological evolution (Meyers et al., 2005). For performing simple analysis, the length of a string is set at 4. According to the setting given in the reference (Meyers et al., 2005), a set of genotypes is defined

Table 1: Set of genotype

Genotype ( $g_i$ )	ID ( $i$ )	Nickname
1011	0	NN1-c
1111	1	NN1-e1
1101	2	NN1-e2
1001	3	NN1-e3
1010	4	NN1-e4
0011	5	NN1-e5
1110	6	INV-1
1000	7	INV-2
0111	8	INV-3
0001	9	INV-4
0100	10	NN2-c
0110	11	NN2-e1
0010	12	NN2-e2
0000	13	NN2-e3
0101	14	NN2-e4
1100	15	NN2-e5

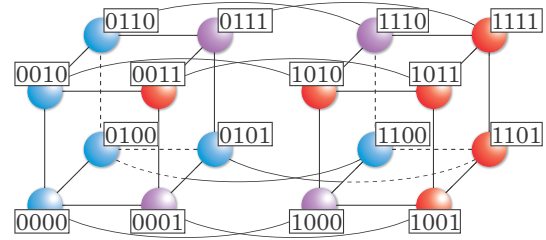


Figure 2: Space of genotype

as Table 1 and Fig. 2. The fitness function is also defined as follows:

$$w_A(g_i) = \begin{cases} 1 + s & (0 \leq i \leq 5) \\ 1 + ks & (6 \leq i \leq 9) \\ 1 & (10 \leq i \leq 15) \end{cases} \quad (3)$$

$$w_B(g_i) = \begin{cases} 1 & (0 \leq i \leq 5) \\ 1 + ks & (6 \leq i \leq 9) \\ 1 + s & (10 \leq i \leq 15), \end{cases} \quad (4)$$

where  $w_A$  and  $w_B$  are fitness functions for environments  $E_A$  and  $E_B$ , respectively.  $s$  and  $k$  ( $s > 0, 0 \leq k \leq 1$ ) are the parameters to adjust the highest and intermediate fitness values given to certain genotypes in each environment, respectively. These parameters were set as follows:  $s = 1, k = 0.5$  following the recommendations given in (Meyers et al., 2005).

In this function, a fitness value is assigned to a genotype directly so no phenotype is defined. Thus, it is considered that  $f_g$  is bijective as mentioned in the previous section. Then  $f_p \circ f_g$  is investigated. According to the definition of a neutral network (Eq.(2)), the genotypes with  $i = 0, \dots, 5$

and those with  $i = 10, \dots, 15$  form a neutral network in both environments,  $E_A$  and  $E_B$ , respectively. These neutral networks show the highest fitness value and lowest fitness value in Eqs. (3) and (4), respectively. In each neutral network, a genotype which does not mutate out of its neutral network with 1 bit is considered to be located at the center of its neutral network (NN1-c and NN2-c in Table 1) while a genotype which does mutate out of its neutral network with 1 bit is considered to be located on the edge of its neutral network (NN1-e and NN2-e in Table 1). For this setting, each neutral network has only one genotype which is located at the center of it. The other genotypes ( $i = 6, \dots, 9$ ) show the intermediate fitness value but do not form any neutral networks.

### Computer Simulation

In this computer simulations, the GA (Goldberg, 1989) were adopted to evolve individuals in both the environments,  $E_A$  and  $E_B$ , mentioned in the previous section. The length of the genotype is 4 as also mentioned in the previous section. The population size was set at 10 according to the setting in (Yu, 2007). In this study, computer simulations were conducted in order to investigate evolutionary dynamics of GAs in a simple model by varying the rates of environmental variation and the mutation rate. Thus, the genetic operations for the GA were standard bit mutation and fitness proportionate reproduction. The per-bit mutation rate,  $q$ , was set as follows:  $q \in \{0.025, 0.05, 0.1, 0.2, 0.25, 0.3, 0.4, 0.5\}$ . Each run lasted 2,000 generations. The initial environment was set at  $E_A$ . The environment was alternately switched every  $\lambda$  generations as follows:  $E_A \rightarrow E_B \rightarrow E_A \rightarrow \dots$ . For each run,  $\lambda$  was set between 1 and 1000 as follows:  $\lambda \in \{1, 2, \dots, 20, 30, \dots, 100, 200, \dots, 1000\}$ . 50 independent runs were conducted for each parameter. All results were averaged over 50 runs.

Fig. 3 shows the ratio of the individuals with the highest fitness value,  $f = 1 + s$ , the intermediate value,  $f = 1 + ks$ , and the lowest value,  $f = 1$  with  $q = 0.025$  and  $\lambda = \{2, 10, 100\}$ <sup>2</sup>. For each  $\lambda$ , a population adapted to a new environment to produce the individuals with the highest fitness value. However, not all individuals converged to the highest fitness value.

The distribution of the individuals were dependent on  $\lambda$ . For short variable periods (e.g.  $\lambda = 2$  in Fig. 3(a)), more than the half of individuals never had the highest fitness value and the individuals with the intermediate fitness value were dominant (approximately 45-50 %). This is because environmental variation was so rapid that there was not enough time for the individuals to adapt to each environment. This might be considered that evolution supported the individuals which can adapt faster to rapid environmental

<sup>2</sup>I plot only the first 100 generations for  $\lambda = 2, 10$  and the first 400 generations for  $\lambda = 100$  because the similar patterns were repeatedly observed after the generations.

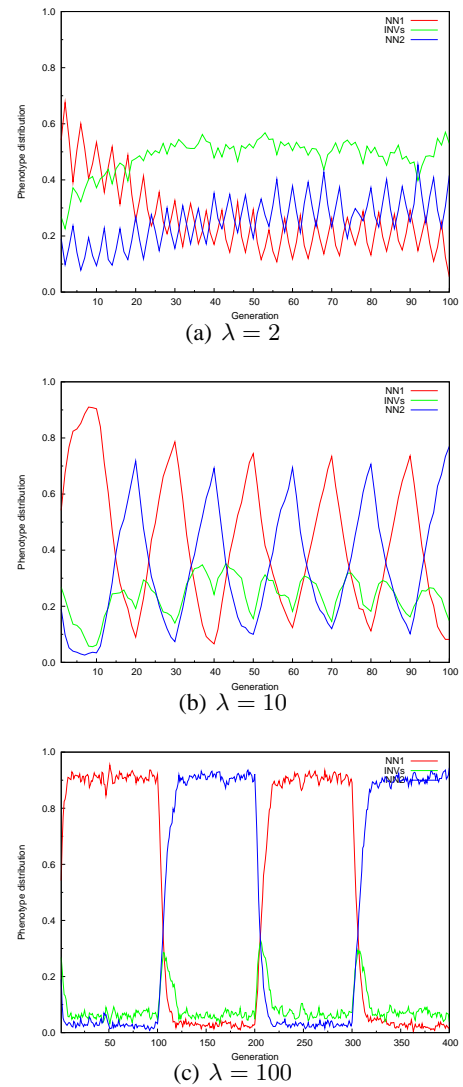


Figure 3: Individual distributions at each generation ( $q = 0.025$ )

variations, that is, the individuals which can mutate easily to the one with the highest fitness value. Such individuals with the intermediate fitness value would be considered to be *organismal flexibility* as mentioned earlier.

For longer variable periods (e.g.  $\lambda = 10$  in Fig. 3(b)), the number of the individuals with the highest fitness value increased while the number of the ones with the intermediate fitness value decreased. For even longer variable periods (e.g.  $\lambda = 100$  in Fig. 3(c)), there was enough time for the individuals to adapt to each environment and the individuals with the highest fitness value became dominant. In Fig. 3, we can not find “where” the individuals are located in the neutral network with the highest fitness. The more details can be found in Figs. 4 and 5.

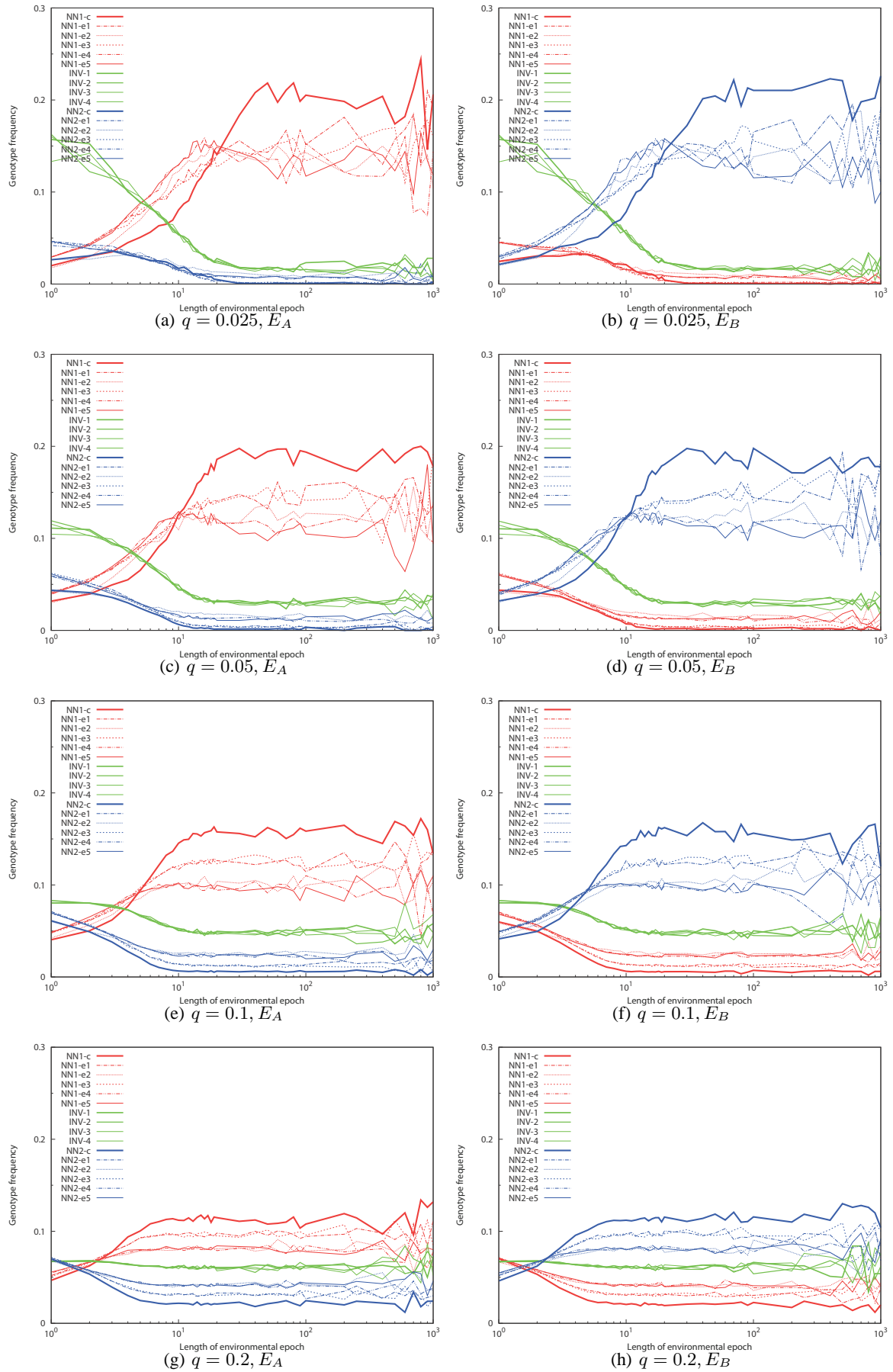


Figure 4: Individual distributions over variable periods for  $E_A$  and  $E_B$  ( $0.025 \leq q \leq 0.2$ )

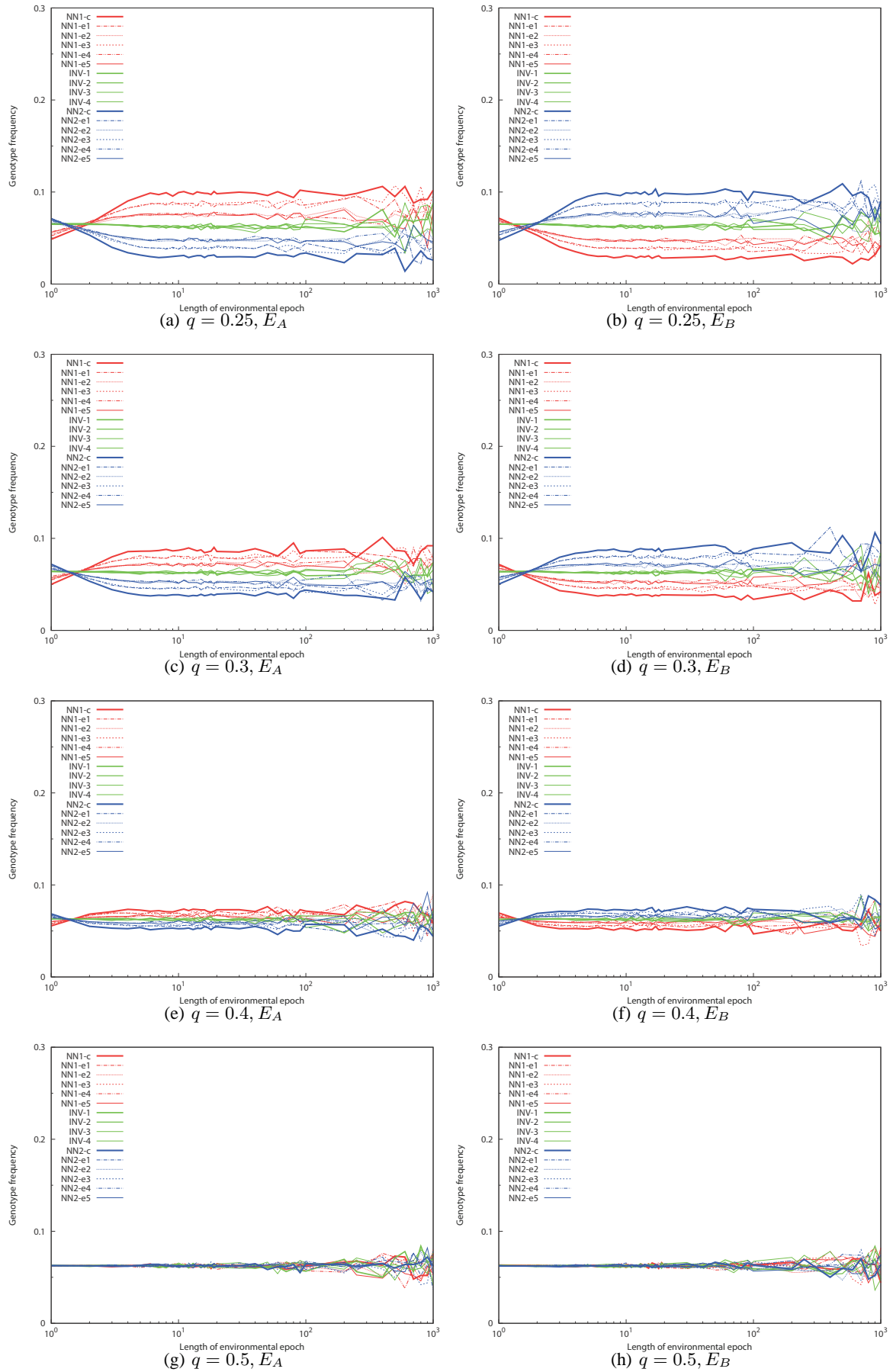


Figure 5: Individual distributions over variable periods for  $E_A$  and  $E_B$  ( $0.25 \leq q \leq 0.5$ )

Figs. 4 and 5 show the ratios of the genotypes over  $\lambda$ s for each  $q$ . Here, the ratio of a genotype was calculated by dividing the sum of the values just before generations when the environment was switched with the number of switching environments and the number of runs. The bold line shows the ratio of the genotype which is located at the center of its neutral network and the thin line shows the one of the genotype which is located on the edge of its neutral network. The horizontal axis is based on a logarithmic scale.

Over the mutation rate range  $0.025 \leq q \leq 0.1$  (Fig. 4(a)-4(f)), for long variable periods, the ratio of the genotype which was located at the center of the neutral network with the highest fitness value was larger than the ones of the other genotypes in both environments. The genotypes which were located on the edge of the neutral network attained the second-largest rate. For shorter variable periods, the ratios of the genotypes which were located on the edge of the neutral network were larger than the ones of the other genotypes. Among them, the ratios were different due to their own locations on the edge. Thus, the ratios of them which are adjacent to not only the genotypes with the intermediate fitness value but also the ones with the lowest fitness value were larger. The variable period range in which this phenomenon appears decreased with the increase of  $q$ . For even shorter variable periods (approximately  $1 \leq \lambda \leq 3$ ), the ratio of the genotype with the intermediate fitness value was largest. For these shortest variable periods, the same result was obtained in Fig. 3(a).

Over the mutation rate range  $0.2 \leq q \leq 0.4$  (Fig. 4(g)-4(h), Fig. 5(a)-5(f)), for long variable periods, the ratios of the genotypes which were located at the center of the neutral network and on the edge of it were large in this order. However, these values were not beyond 0.1. For even shorter variable periods, the ratios of the genotypes with the lowest fitness value were a few larger than or equal to the ones with the highest and intermediate fitness value.

For  $q = 0.5$  (Fig. 5(g)-5(h)), there was no significant difference between the genotypes, which were distributed randomly.

## Discussion

In the earlier section, the loosely defined phenomena, *genetic robustness*, *genetic potential* and *organismal flexibility*, were cited. In order to discuss the results obtained in the previous section, those are more accurately defined as follows: *genetic robustness*: the state where the ratio of the genotype which is located at the center of the neutral network with the highest fitness value is largest in the environment,  $E_A$  or  $E_B$ . *genetic potential*: the state where the ratio of the genotype which is located on the edge of the neutral network is largest in each environment. *organismal flexibility*: the state where the ratio of the genotype with the intermediate fitness value is largest.

According to these definitions, we can find such phase

transitions as organismal flexibility  $\rightarrow$  genetic potential  $\rightarrow$  genetic robustness for  $q \leq 0.1$ , and organismal flexibility  $\rightarrow$  genetic robustness for  $0.2 \leq q \leq 0.4$  in Figs. 4 and 5 with the increase of the variable period. Meyers et al. (2005) described that we can find *genetic potential* in a much wider variable period range when the mutation rate decreases because it takes more time to reach the state, *genetic robustness*. This is consistent with the results in Figs. 4(a)-4(f) in which we can find *genetic potential* in the ranges,  $6 \leq \lambda \leq 20$  for  $q = 0.025$ ,  $5 \leq \lambda \leq 10$  for  $q = 0.05$  and,  $3 \leq \lambda \leq 5$  for  $q = 0.1$ . Meyers et al. did not mention *organismal flexibility* for the high mutation rates. For the results obtained in this study, we can not find any *organismal flexibility* when the mutation rate was high. This would be considered to be affected by the *error threshold* on the mutation rate (Kauffman, 1995); As the mutation rate increases, the population gradually loses the current individuals. At a certain critical mutation rate, the individuals become distributed randomly.

Meyers et al. (2005) also claimed that the mutation rate per locus does not need to be variable if the phenotypical mutation rate or the effective mutation rate per genotype is variable as opposed to the argument that the variable mutation rate per locus is important for adaptation to environmental variations. This argument would be explained as follows. When the mutation rate per locus is low, individuals must change their phenotypes (or obtain the higher fitness value) as soon as possible in order to adapt to environmental variation. Thus, the individuals which are located on the edge of the neutral network are supported. When the mutation rate per locus is high, individuals can change quickly their phenotypes even though they are located at the center of the neutral network. Therefore, the dominance of the individuals which are located on the edge of the neutral network becomes invisible at such a mutation rate.

## Conclusions

This study investigated evolutionary dynamics of GAs in a simple model by varying the rates of environmental variation and the mutation rate. The results can be summarized as follows:

- Two or three phase transitions were observed over the variable period range. Especially when the mutation rate is low, the results were consistent with the results obtained in computational biology.
- For long variable periods, the frequency of the genotype which was located at the center of the neutral network with the highest fitness value was largest in the population.
- For shorter variable periods, the frequency of the genotype which was located on the edge of the neutral network was largest.

- For even shorter variable periods, the frequency of the genotype with the intermediate fitness value was largest.

In this study, four-bit binary strings were used to provide simple explanatory examples. Additionally, a small population size and an alternating environment were set. Further computer simulations will be conducted in order to investigate whether these observations are consistent with more complex settings (Yang et al., 2007). Another future direction would be an analytical approach due to the simplicity of the model.

## Acknowledgements

This work was supported by Grants-in-Aid for Scientific Research 22700241.

## References

- Barnett, L. (1997). Tangled webs: Evolutionary dynamics on fitness landscapes with neutrality. Master's thesis, School of Cognitive and Computing Sciences, Sussex University, Brighton, UK.
- Branke, J. (2001). *Evolutionary Optimization in Dynamic Environments*. Kluwer Academic Publishers.
- Ebner, M., Langguth, P., Albert, J., Shackleton, M., and Shipman, R. (2001). On neutral networks and evolvability. In *Proceedings of the 2001 IEEE Congress on Evolutionary Computation: CEC2001*, pages 1–8, Piscataway, New Jersey. IEEE Press.
- Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.
- Harvey, I. (1997). Artificial evolution for real problems. In Gomi, T., editor, *Evolutionary Robotics: From Intelligent Robots to Artificial Life (ER'97)*. AAI Books, Tokyo.
- Harvey, I. and Thompson, A. (1996). Through the labyrinth evolution finds a way: A silicon ridge. In *Proceedings of the first International Conference on Evolvable Systems: From Biology to Hardware*, pages 406–422.
- Katada, Y. and Ohkura, K. (2006). Estimating the degree of neutrality in fitness landscapes by the nei's standard genetic distance – an application to evolutionary robotics –. In *Proceedings of the 2006 IEEE Congress on Evolutionary Computation (CEC2006)*, pages 1590–1597.
- Katada, Y. and Ohkura, K. (2009). Analysis on topologies of fitness landscapes with both neutrality and ruggedness based on neutral networks. In *Proceedings of the 2009 Genetic and Evolutionary Computation Conference (GECCO2009)*, pages 1855–1856.
- Katada, Y., Ohkura, K., and Ueda, K. (2004). An approach to evolutionary robotics using the genetic algorithm with variable mutation rate strategy. In *Proceedings of The 8th Parallel Problem Solving from Nature (PPSN VIII)*, pages 952–961.
- Kauffman, S. A. (1995). *At Home in the Universe: The Search for Laws of Self-organization and Complexity*. Oxford University Press.
- Kimura, M. (1983). *The Neutral Theory of Molecular Evolution*. Cambridge University Press, New York.
- Knowles, J. D. and Watson, R. A. (2002). On the utility of redundant encodings in mutation-based evolutionary search. In Merelo, J., Admidis, P., Beyer, H.-G., Fernandes-Villacanas, J.-L., and Schwefel, H.-P., editors, *Proceedings of Parallel Problem Solving from Nature - PPSN VII, Seventh International Conference*, pages 88–98, Granada, Spain. LNCS 2439.
- Meyers, L., Ance, F., and Lachmann, M. (2005). Evolution of genetic potential. *Computational Biology*, 1(3):236–243.
- Miller, J. (2009). Cartesian genetic programming. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation Conference, Tutorial*, pages 3489–3512.
- Newman, M. and Engelhardt, R. (1998). Effect of neutral selection on the evolution of molecular species. In *Proc. R. Soc. London B.*, pages 256:1333–1338.
- Ohkura, K. and Ueda, K. (1999). Adaptation in dynamic environment by using GA with neutral mutations. *International Journal of Smart Engineering System Design*, 2:17–31.
- Rothlauf, F. and Goldberg, D. E. (2003). Redundant representations in evolutionary computation. *Evolutionary Computation*, 11(4):381–415.
- Smith, T., Husbands, P., Layzell, P., and O'Shea, M. (2002). Fitness landscapes and evolvability. *Evolutionary Computation*, 10(1):1–34.
- Smith, T., Husbands, P., and O'Shea, M. (2001). Neutral networks in an evolutionary robotics search space. In *Proceedings of the 2001 IEEE Congress on Evolutionary Computation: CEC2001*, pages 136–145, Piscataway, New Jersey. IEEE Press.
- Thompson, A. (1996). An evolved circuit, intrinsic in silicon, entwined with physics. In *Proceedings of the first International Conference on Evolvable Systems: From Biology to Hardware*, pages 390–405.
- van Nimwegen, E., Crutchfield, J. P., and Mitchell, M. (1999). Statistical dynamics of the royal road genetic algorithm. *Theoretical Computer Science*, 229(1):41–102.
- Vanneschi, L. (2009). Fitness landscapes and problem fairness in genetic programming. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation Conference, Tutorial*, pages 3657–3684.
- Vassilev, V. K., Miller, J. F., and Fogarty, T. C. (2000). Information characteristics and the structure of landscapes. *Evolutionary Computation*, 8(1):31–60.
- Yang, S., Ong, Y.-S., and Jin, Y., editors (2007). *Evolutionary Computation in Dynamic and Uncertain Environments*. Springer.
- Yu, T. (2007). Program evolvability under environmental variations and neutrality. In *Proc. ECAL 2007*, pages 835–844. LNAI 4648.
- Yu, T. and Miller, J. F. (2006). Through the interaction of neutral and adaptive mutations, evolutionary search finds a way. *Artificial Life*, 12(4):525–551.



# Why Aren't Protein Residue Networks Smaller Worlds

Susan Khor<sup>1</sup>

<sup>1</sup>Montréal, Canada  
slc.khor@gmail.com

## Abstract

In this paper, *in silico* experiments are performed to investigate why protein residue networks (PRNs), i.e. networks induced by spatial contacts between amino acid residues of a protein, do not have shorter average path lengths (APLs) in spite of their importance to protein folding. We find that shorter average inter-nodal distances does not necessarily imply better search performance, i.e. more successful protein folding. Search performance of a zero-temperature Metropolis style hill-climber was not significantly improved by randomizing only the long-range links of PRNs even though such randomization significantly reduces APLs of PRNs while retaining much of the clustering and positive degree-degree correlation inherent in PRNs. However, this result is contingent upon the optimization function. We found that the optimization function which places PRNs in a favorable spot in the space of possible network configurations considered in this paper parallels an existing view in protein folding theory that neither short-range nor long-range interactions dominate the protein folding process. These findings suggest the existence of explanations, other than the excluded volume argument, beneath the topological limits of PRNs.<sup>1</sup>

## Introduction

Breaking the code underlying protein folding has remained an intellectually tantalizing puzzle as well as a problem of great practical significance. Everything a protein requires for correct folding under normal circumstances appears to be embedded in its amino-acid sequence (Afinsen 1973), although a minority rely on the aid of water and chaperone molecules. Due to the large sizes that amino-acid sequences can take, a random search approach to protein folding is deemed infeasible for practical biological purposes (Levinthal 1969). However, an argument based on separability of the protein folding problem, i.e. that the problem can be separated into parts which can then be solved independently and assembled into an optimal solution<sup>2</sup>, has been conceived as a way out of Levinthal's paradox (Zwanzig *et al* 1992; Karplus 1997). This argument is supported by some sections of protein

sequences having a propensity to fold to their native secondary structures.

In general, protein folding is a process that occurs in stages. What essentially begins as a linear hetero-polymer (organized as a backbone with protruding side chain groups) obtains local structure in the form of secondary alpha helices and beta sheets and finally global structure as the secondary structures arrange themselves compactly in three dimensions. For a long time, this spontaneous biological self-organization has been attributed to various inter-atomic physical forces and chemical constraints impacting a protein molecule. However, in the last decade or so, another theory based on the network topology of a protein's native state has blossomed. In this other theory, a network view of protein molecules (mostly in their native states) is adopted.

The general recipe to transform a protein molecule into a network is to represent amino acid residues ( $C_\alpha$  or  $C_\beta$ ) as nodes, and contact (spatial, non-covalent) distances between pairs of amino acid residues below a certain threshold as links. Such *protein residue networks* (PRN) are constructed from the Cartesian coordinates of amino acid residues of protein molecules stored in the Protein Data Bank (PDB) (Berman *et al* 2000). There are variations to the general recipe however. For instance, a PRN may represent several non-homologous proteins rather than a single protein, e.g. the protein contact map in (Vendruscolo *et al* 2002). PRNs may also represent different aspects (e.g. surface or core), states (e.g. native or transitional), structural classes (e.g.  $\alpha$ ,  $\beta$ ,  $\alpha+\beta$  or  $\alpha/\beta$ ), or types (e.g. globular or fibrous) of proteins (e.g. Atilgan *et al* 2004). Further, the nodes and links of PRNs may carry different meanings, e.g. the atoms of the side chain group of an amino acid may be included so that a node may represent more than one atom and multiple links between nodes or weighted links are allowed (Green and Higman 2003).

By examining PRNs, researchers have compiled a list of topological characteristics shared by a diverse (in terms of structural class, homology and taxon) set of proteins and speculated on the reasons for the observed topological characteristics in relation to protein folding. A common feature of protein residue networks is their *small-world* nature, i.e. they have lattice like clustering coefficients but random graph like diameters and average path lengths (Watts and Strogatz 1998). The need for rapid communication between amino acid residues to facilitate *interaction cooperativity* crucial for protein folding is frequently cited as the reason for the small-world feature of PRNs (Vendruscolo *et al* 2002; Dokholyan *et al* 2002; Atilgan *et al* 2004; Del Sol *et al* 2006). PRNs are also reported to exhibit high assortativity values

<sup>1</sup> This is an independent research paper, part of which was completed during the author's stay at Collegium Budapest, Hungary who generously provided the computer resources for most of the experiments. At the time this paper was prepared, the author is a post-doctoral researcher in Montréal, Canada.

<sup>2</sup> For a more colourful description, see Herbert Simon's parable of the two watchmakers in *The Sciences of the Artificial*, 1969 MIT Press.

which can be related to protein folding speeds (Bagler and Sinha 2007). We discuss network characteristics of PRNs further on.

In this paper, we set out to understand why, given the assumed importance of rapid inter-residue communication to protein folding, PRNs are not smaller-worlds or equivalently do not have shorter average shortest path lengths. We address this question from a search perspective, which is not unusual given the common formulation of protein folding as a search problem. We define a spin-glass like problem on a PRN and use the performance of a local search (a hill-climber in the fashion of the Metropolis algorithm with zero probability of assuming a higher temperature configuration) to assess the effect of changes in network topology, specifically average path length (APL), on search performance. The experiments are conducted under several conditions, motivated by existing literature on protein folding theory.

## Method

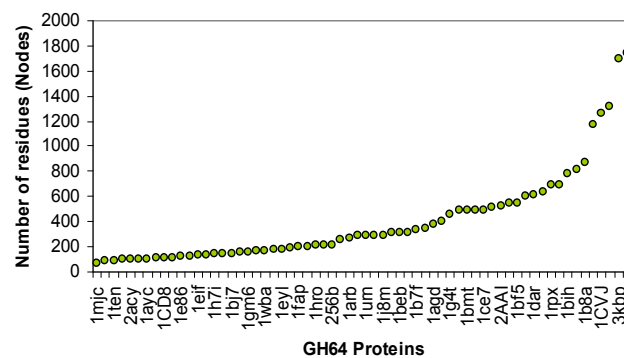
### Protein residue network construction

Our PRN has  $N$  nodes (one for each amino acid of a protein) and  $M$  links. An undirected link is placed between a pair of nodes representing the  $C\alpha$  atom of amino acids when the node pair is situated less than  $7\text{\AA}$  apart from each other. The small-world property of PRNs is not overly sensitive to the choice of this threshold value (Bartoli *et al* 2007). Distance between node pairs is the Euclidean distance between their 3D Cartesian coordinates obtained from the Protein Data Bank or PDB (Berman *et al* 2000). The  $M$  links are partitioned into two sets: long-range links ( $LE$ ) and short-range links ( $SE$ ). A link between nodes  $x$  and  $y$  is classified as *long-range* if their absolute distance on the amino acid sequence chain is more than 9 (Green and Higman 2003). Long-range links connect amino acids which are distant in the primary structure but are in close spatial proximity in the tertiary structure.

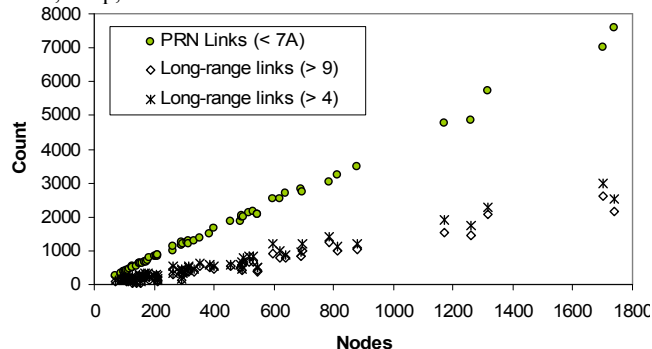
### Test data set

A PRN is built for each protein in the GH64 dataset (Figures 1&2) which was selected from literature surveyed, specifically (Green and Higman 2003). The dataset encompass proteins from different protein classes, fold types and branches of life. Proteins which did not form a single connected component (i.e. 1cuk and 1ho4), or had unusually high link density (i.e. 1feo) in its PRN were excluded from the dataset. So too were proteins with more nodes in their PRN than their DSSP output (Kabsch and Sander 1983) (i.e. 2hmz and 1epf). We use the output from DSSP (Dictionary of Protein Secondary Structure) as globally optimal strings in our search problem. If the reverse situation occurs, the DSSP output is truncated. A second dataset, EVA132, is used to increase confidence of key results in this paper. The EVA132 protein dataset was extracted from the list of 3477 unique chains archived by EVA (Rost, 1999). 200 proteins were selected at random from this list, with no overlap with GH64. PRNs for these 200 proteins were constructed and selected in the same manner as GH64, yielding 132 well-formed PRNs. EVA132 PRNs possess similar network characteristics as

GH64 PRNs. Detailed information on both sets can be found in <http://arxiv.org/abs/1011.2222>.



**Figure 1** Size of GH64 proteins in terms of the number of  $C\alpha$  atoms. PIDs are: 1mjc, 1gvp, 1ten, 1ris, 2acy, 1tlk, 1ayc, 1sha, 1CD8, 1d4t, 1e86, 2fgf, 1eif, 1pdo, 1h7i, 1amx, 1bj7, 1aep, 1gm6, 3rab, 1wba, 1rbp, 1eyl, 153L, 1fap, 1nsj, 1hro, 1jr8, 256b, 1ICE, 1arb, 1vlt, 1urn, 1amp, 1j8m, 1cjl, 1beb, 1OBP, 1b7f, 1hng, 1agd, 1aye, 1g4t, 1eov, 1bmt, 7tim, 1ce7, 1hwn, 2AAI, 1fbv, 1bf5, 1jly, 1dar, 1eun, 1rpx, 1bbp, 1bih, 1psd, 1b8a, 1ava, 1CVJ, 3eca, 3kbp, 1dio.



**Figure 2** Link count  $M$ , by protein size for GH64 PRNs.

### Search problem and search algorithm

We define a spin-glass like problem on a PRN and use the performance of a local search (a hill-climber in the fashion of the Metropolis algorithm with zero probability of assuming a higher temperature configuration) to assess the effect of changes in network topology, specifically average path length, on search performance. Starting at random points in a search space comprising  $\{0, 1, 2\}^N$  strings<sup>3</sup>, the problem is to find  $s$ , the unique globally optimal string defined by the DSSP output (Kabsch and Sander 1983) for a PRN reduced with the following rules: 0 represents H, I, and G, 1 represents E and B, and 2 represents others. The unique global optimum is reachable by maximizing the following fitness function which is derived from (Bryngelson and Wolynes, 1987):

$$\sum_{i=0}^N g(s_i, \mathbf{s}_i) + \sum_{i=0}^M f(e_i, s, \mathbf{s}).$$
 Define  $s_i$  as the current value of the  $i^{\text{th}}$  element in string  $s$ .  $g(s_i, \mathbf{s}_i) = 1$  if  $s_i = \mathbf{s}_i$  and 0 otherwise.

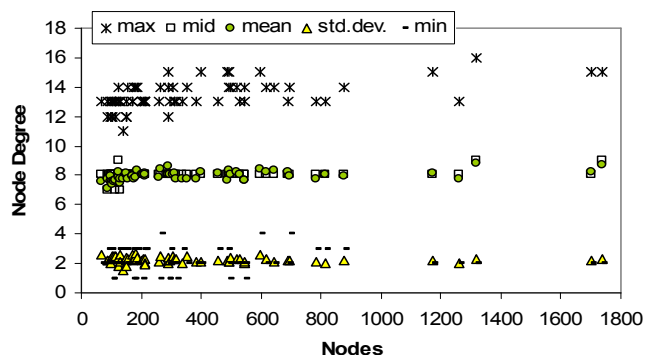
<sup>3</sup> Incidentally,  $3^N$  search spaces are common in discrete models of protein folding, e.g. 3 possible peptide bond torsion angles, and 3 possible bonds between hydrophobic (H) and polar (P) residues.

Define  $e_i$  as the  $i^{\text{th}}$  link in a PRN and  $e_i$  connects nodes  $j$  and  $k$ .  $f(e_i, s, \mathbf{s}) = 1$  if  $|s_j - s_k| = |\mathbf{s}_j - \mathbf{s}_k|$  and 0 otherwise. The  $g$  term ensures a unique global optimum<sup>4</sup> while the  $f$  term introduces frustration, i.e. the required ruggedness feature into the fitness landscape (Dill *et al* 1995, p.585).

The local search algorithm is a hill climber which at each time step, the value of a single randomly chosen element assumes a different value chosen randomly from  $\{0, 1, 2\}$ , and never moves down hill to less fit points. For each run, the hill climbing algorithm is iterated until  $\mathbf{s}$  is found, or the fitness function has been evaluated 1 million times. 20 independent runs are made per PRN. A total of 1280 (64 x 20) and 2640 (132 x 20) runs are made for GH64 and EVA132 respectively.

## Network Characteristics of PRNs

**Node degree** measures the number of contacts or direct neighbors a node has in a PRN. Gaci and Balev (2009) remarked on the homogeneity of node degree in their PRN called SSE-IN which only considers secondary structure elements. The mean node degree of their SSE-INS increased very slightly with protein size and fell within the range of 5 and 8. The absence of nodes with much higher degrees is attributed to the excluded volume effect which imposes a physical limit on the number of residues that can reside within a given radius around another amino acid. The mean node degree ( $K$ ) of the GH64 PRNs averages at 7.9696 with a standard deviation of 0.3126, and is independent of protein size (Figure 3).



**Figure 3** Node degree summary statistics for GH64 PRNs.

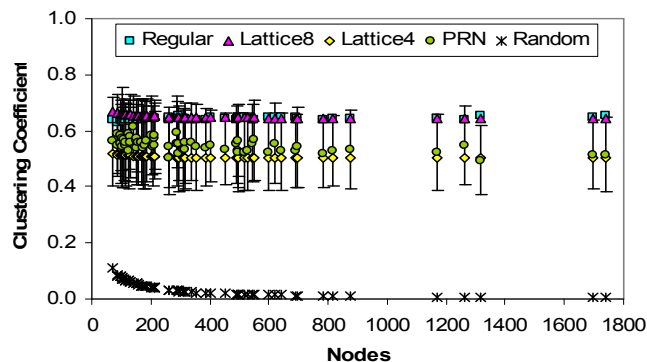
**Clustering** or transitivity reflects the cliquishness of nodes in a network: if node  $X$  connects to node  $Y$  and to node  $Z$ , how likely is it that nodes  $Y$  and  $Z$  are connected to each other? A convenient way to measure network clustering is by taking the average clustering of all nodes in a network to yield

the clustering coefficient as follows:  $C = \frac{1}{N} \sum_{i=1}^N \frac{2e_i}{k_i(k_i - 1)}$

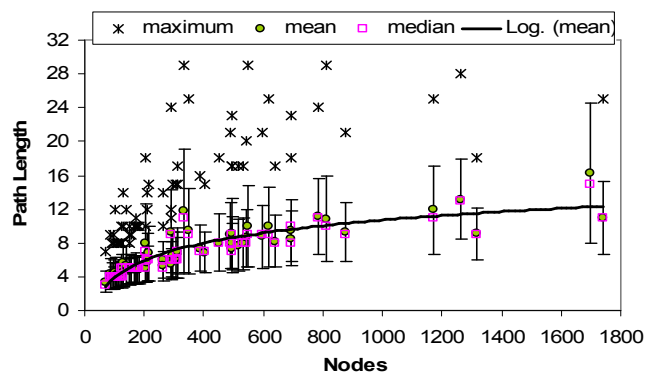
where  $k_i$  is the degree of node  $i$ , and  $e_i$  is the number of links that exist amongst the  $k_i$  nodes (Watts and Strogatz 1998). Independent of protein size, the  $C$  values for GH64 PRNs ( $C_{\text{GH64}}$ ) are significantly higher than  $C_{\text{RANDOM}}$ , and closest to

<sup>4</sup> It also has a separable or a smoothing effect; without the  $f$  term, there is no interaction between variables.

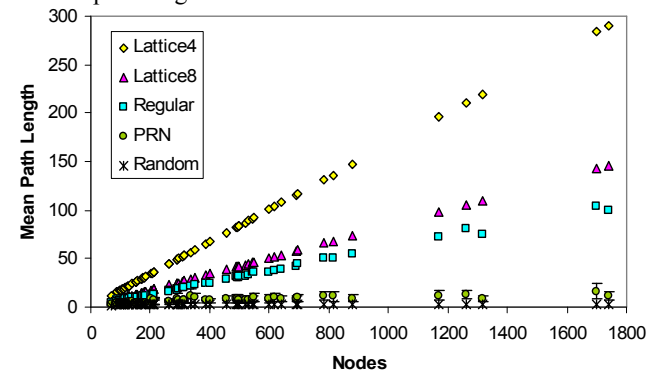
$C_{\text{LATTICE4}}$  (Figure 4). Lattice $V$  is a linear lattice with  $V/2$  nearest neighbours to the left and to the right where possible.



**Figure 4**  $C_{\text{GH64}}$  values ( $\pm$  one standard deviation) compared with  $C_{\text{Lattice8}}$ ,  $C_{\text{Lattice4}}$  and the theoretical  $C$  values for regular ( $C_{\text{REGULAR}}$ ) and random ( $C_{\text{RANDOM}}$ ) networks of the same size (same number of nodes).  $C_{\text{RANDOM}} \sim K / N$ , and  $C_{\text{REGULAR}} = 3(K-2) / [4(K-1)]$ , where  $K$  is average degree and  $N$  is number of nodes (Watts 1999). We use  $K=8$  (see Figure3).



**Figure 5** Diameter, average ( $\pm$  one standard deviation) and median path length for GH64 PRNs.



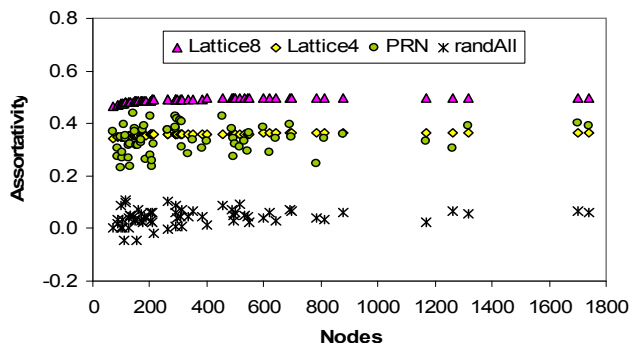
**Figure 6** APLs of PRNs are much closer to APLs of random networks ( $\text{APL}_{\text{RANDOM}}$ ) than to APLs of regular networks ( $\text{APL}_{\text{REGULAR}}$ ).  $\text{APL}_{\text{RANDOM}} \sim \ln N / \ln K$ , and  $\text{APL}_{\text{REGULAR}} = N(N + K - 2) / [2K(N - 1)]$ , where  $K$  is average degree and  $N$  is number of nodes (Watts 1999). We use  $K=8$  (see Figure3).

The **average path length** (APL) of a network is the average length of a set of shortest paths between all node-pairs. The average path length for GH64 PRNs ( $\text{APL}_{\text{GH64}}$ ) increases logarithmically with increases in protein size (nodes) (Figure 5). When compared with average path lengths

of other canonical networks,  $APL_{GH64}$  is much shorter than the average path lengths of regular graphs ( $APL_{REGULAR}$ ) and approximate the average path lengths expected of random graphs ( $APL_{RANDOM}$ ) of the same size (Figure 6).  $APL_{GH64}$  is also much shorter than both  $APL_{LATTICE8}$  and  $APL_{LATTICE4}$  (Figure 6).

The **small-world property** is a combination of high clustering and short inter-nodal distances (average path length increases logarithmically with network size), two conditions that from the above exposition, GH64 PRNs satisfy.

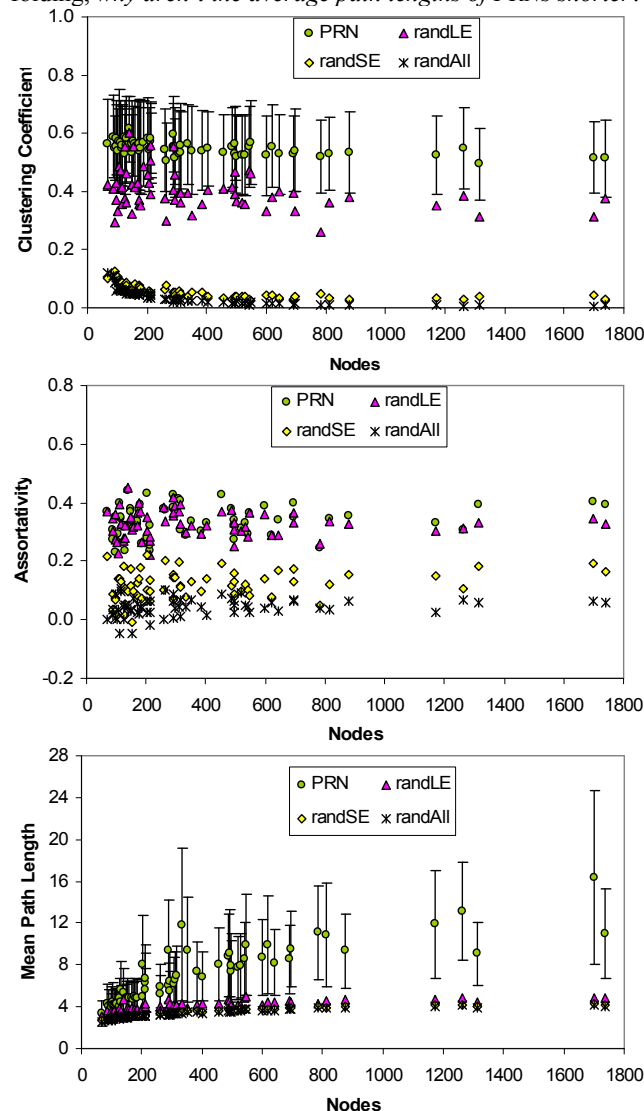
**Assortativity** refers to the extent that nodes associate or connect with their own kind. A common form of assortativity measured for PRNs is node degree. Positive degree-degree assortativity refers to the proclivity of nodes with small (large) degree to link with other nodes of small (large) degree. Using the method in (Newman 2002), Bagler and Sinha (2007) report degree-degree correlation coefficients up to 0.58, which is considered unusual for networks with biological origins. Nonetheless, the positive assortativity values could be correlated in a positive manner to protein folding speeds (Bagler and Sinha 2007). Similarly, we find positive degree-degree correlations in the GH64 PRNs independent of protein size. The assortativity values average at 0.3387 with a standard deviation of 0.0536, which is much higher than observed for randomized PRNs (randAll) (Figure 7). For randAll networks, PRNs are randomized in the usual manner by rewiring nodes while preserving node degrees and without introducing multiple links between nodes (Maslov and Sneppen 2002).



**Figure 7** The GH64 PRNs have positive degree assortativity, with values closest to those for Lattice4.

As with clustering (Figure 4), the assortativity values for GH64 PRNs are closest to Lattice4 (Figure 7). Bartoli *et al* (2007) commented that links encompassing a protein's backbone (which are short-ranged) is the main source of the relatively high levels of clustering in PRNs. We observed that short-range links (SE) are also responsible for much of the positive assortativity in PRNs. Figure 8 shows the effect of randomizing different sets of links in GH64 PRNs. Both clustering and assortativity levels show larger decreases when only the short-range links are randomized (randSE), compared with when only the long-range links are randomized (randLE). The APLs of PRNs are significantly reduced in both randSE and randLE networks. Hence it is possible, by randomizing only the long-range links, to rearrange the links of a PRN such that the APL is significantly reduced while preserving clustering and positive degree-degree assortativity coefficients at levels higher than would be in random graphs. Our question

is thus: if short inter-nodal distances are important for protein folding, *why aren't the average path lengths of PRNs shorter?*



**Figure 8** Effect of randomizing different sets of PRN links on Clustering Coefficient (top), Assortativity (middle), and Average Path Length (bottom). Error bars denote one standard deviation from the mean.

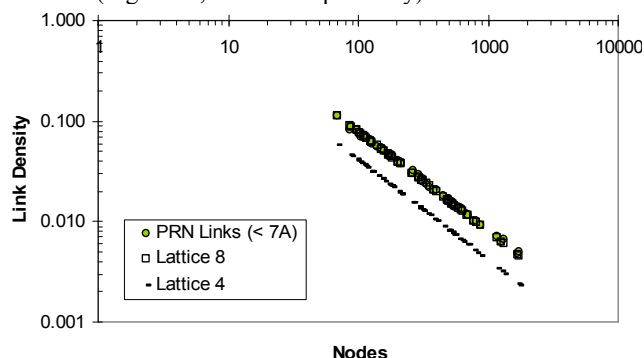
## Results and Discussion

Both accuracy and speed, i.e. finding the right structure consistently in biologically functional time, are important criteria in the protein folding problem. We measure accuracy of the local search in terms of *Success Rate (SR)*, which is the proportion of total runs (20) per PRN where the hill climber found the unique global optimum within 1 million evaluations. Speed of the local search is accessed by *avg\_evals*, which is the number of fitness function evaluations averaged over all runs with  $SR > 0.0$  per PRN. Configuration *A* is considered more favorable to protein folding than configuration *B* if the local search algorithm performs better, i.e. achieves a significantly higher SR and a significantly

lower avg\_evals, on *A* than on *B*. A *configuration* refers to a combination of network topology and fitness function. Search performance is affected by network size, larger networks are in general expected to be either more difficult to optimize and/or require more function evaluations. To remove this size effect, search performances between two configurations are compared on the set of common networks with SR > 0.0. The largest p-value of the Shapiro-Wilk test for SR and avg\_evals data is 0.03380715 and 1.072698e-07 respectively. This allows us to conclude, with at least 95% confidence, that both SR and avg\_evals data are not normally distributed, and to use the Wilcoxon method (paired) to test SR and avg\_evals data for significance. Following this procedure, the hypotheses in the following discussion are confirmed with at least 95% confidence.

## Experiment 1

The objective is to compare PRNs with lattices (“regular” graphs), i.e. how differences in their network topology affect search performance given the fitness function stated earlier. Lattice $V$  is a linear lattice with  $V/2$  nearest neighbours to the left and to the right where possible ( $V/2$  nodes at each of the two ends of the lattice chain will have fewer links than the rest of the nodes in the middle which will have  $V$  links each). The GH64 PRNs share several network characteristics with Lattice4 and Lattice8. For networks having the same number of nodes, PRNs have the same link density as Lattice8, and similar clustering and positive degree-degree assortativity levels as Lattice4 (Figures 9, 4 and 7 respectively).



**Figure 9** Link density by size for GH64 proteins on a log-log plot, compared with the same for Lattice4 and Lattice8. Link density is the fraction of actual links out of all possible links, i.e.  $2M / [N(N-1)]$ .

Using the fitness function defined earlier, the hill climber performed better when the network topology is PRN than when it is Lattice8 (Table 1). However, both are outperformed by Lattice4 (Table 1), which has a significantly longer APL (Figure 6). Lattice4 also produced twice as many networks with SR > 0.5, and seems more conducive to larger networks than either Lattice8 or PRN (Table 2). Hence, shorter inter-nodal distances do not guarantee better search performances. The shorter APLs of Lattice8 and of PRN are the result of more links (Figure 9), and the fitness function is such that additional links can increase frustration (more on this point in Experiment 4). Furthermore, the regular connectivity of a lattice network probably does not produce suitably functional surface structures like those of proteins (but see Table 7).

**Table 1** Result summary for Experiment 1

A	B	#	SR	avg_evals
Lattice8	<b>PRN</b>	49	A = B	A > B
<b>Lattice4</b>	Lattice8	53	A > B	A < B
<b>Lattice4</b>	PRN	56	A > B	A < B

# is the number of networks with SR > 0.0, common to both configurations *A* and *B*. Configurations with better overall search performances are **bolded**. These signs apply to all subsequent result tables. PRNs are from the GH64 dataset. Optimal strings for PRN and both Lattice4 and Lattice8 come from the DSSP output for proteins in the GH64 dataset (as explained earlier in the Method section).

## Experiment 2

The objective is to observe the effect of link randomization on search performance. By randomizing different sets of links in a PRN, it is possible to rearrange the links of a PRN such that the APL is significantly reduced (Figure 8). Further, by randomizing only the long-range links (randLE), it is also possible to significantly reduce APLs of PRNs and maintain clustering coefficients and positive degree-degree assortativity at levels higher than would be in random graphs (Figure 8).

**Table 3** Result Summary for Experiment 2

A	B	#	SR	avg_evals
randSE	randAll	64	A = B	A = B
PRN	<b>randLE</b>	55	A < B	A > B
PRN*	<b>randLE*</b>	115	A < B	A > B

PRNs are from the GH64 and where indicated by \*, the EVA132 dataset. randLE, randSE and randAll are as explained earlier, PRN networks produced by respectively randomizing only the Long-range links, only the Short-range links and all links. Optimal strings for all networks come from the DSSP output for GH64 proteins and where indicated by \*, the DSSP output for EVA132 proteins.

Compared to PRN, randomizing all links (randAll) and randomizing short-range links (randSE) increased SR to almost 100%, with a considerable decrease in avg\_evals (Table 2). There is no significant difference in terms of search performances between randSE and randAll (Table 3). But randAll and randSE networks lose much of the local organizational structure of PRN networks (Figure 8), and so probably do not produce suitably functional surface structures like those of proteins (see Table 7). *What is more interesting is that randomizing long-range links (randLE) significantly improved search performance over PRN (Table 3) for both GH64 and EVA132 datasets.* We revisit this point in Experiment 4, where an adjustment to the fitness function restores PRN to a favourable spot in the space of possible network configurations considered in this paper.

## Experiment 3

The objective is to compare the relative importance of short-range and long-range links to protein folding. There has been quite an evolution of thought in this area (Dill et al, 1995; Gö, 1983) and is by no means a settled issue. All three possible perspectives have been contemplated: (i) primacy of short-range interactions, (ii) primacy of long-range interactions, and (iii) non-dominance of either set of interactions.



**Table 2** Key summary statistics for results obtained in Experiments 1, 2 and 3

PRN s	Configuration	Median, Mean, Sd Success Rate (SR)	Proportion of networks with > 0.0 SR	Median avg_evals of networks with > 0.0 SR	Proportion of networks with > 0.5 SR	Median size of networks with > 0.5 SR
GH64	Lattice4	0.6000, 0.5727, 0.3142	62/64 = 0.9688	6158	35/64 = 0.5469	211
	Lattice8	0.2500, 0.3289, 0.3190	53/64 = 0.8281	7245	15/64 = 0.2344	185
	PRN	0.3250, 0.3383, 0.2569	56/64 = 0.8750	6264	14/64 = 0.2188	129
	randLE	0.3750, 0.4000, 0.2772	62/64 = 0.9688	5126	24/64 = 0.3750	134
	randSE	1.0000, 0.9977, 0.0139	64/64 = 1.0000	3609	64/64 = 1.0000	290
	randAll	1.0000, 0.9945, 0.0220	64/64 = 1.0000	3541	64/64 = 1.0000	290
	onlySE	0.0500, 0.1313, 0.2124	33/64 = 0.5156	4115	6/64 = 0.0938	134
	onlyLE	1.0000, 0.9070, 0.1466	64/64 = 1.0000	4262	62/64 = 0.9688	276
	delay07	0.3500, 0.3734, 0.2619	58/64 = 0.9063	5703	20/64 = 0.3125	146
	delay08	0.3500, 0.3703, 0.2735	57/64 = 0.8906	5926	18/64 = 0.2813	146
	delay09	0.3500, 0.3789, 0.2823	56/64 = 0.8750	5733	21/64 = 0.3281	153
	delay10	0.0500, 0.1273, 0.2074	33/64 = 0.5156	4115	6/64 = 0.0938	134
EVA132	PRN	0.2500, 0.3008, 0.2460	119/132 = 0.9015	9976	27/132 = 0.2045	145
	randLE	0.3500, 0.3553, 0.2508	120/132 = 0.9091	10580	33/132 = 0.2500	226

The first column gives the source of the PRN and optimal string *s*. Size of networks in the GH64 dataset has a median of 290 and is not normally distributed. The one-sample Kolmogorov-Smirnov two-sided test *p*-value = 2.220e-16. Size of networks in the EVA132 dataset has a median of 442 and is not normally distributed. The one-sample Kolmogorov-Smirnov two-sided test *p*-value is < 2.2e-16.

Compared with PRN, the use of only short-range links to guide the search (onlySE) reduced SR by 41% while using only long-range links (onlyLE) increased SR by 14% to almost 100% (Table 2). By examining both GH64 and EVA132 PRNs, we found that on average, only about 30% of all links in a PRN are long-range. The satisfaction of all links (constraints) in a PRN is necessary to achieve the globally maximal fitness value, and perfectly relaxed protein molecules as described by Gō (1983). Thus, the SR for onlyLE is quite remarkable and lends credence to the “primacy of long-range interactions” view (Dill *et al.*, 1995). In both GH64 and EVA132 PRNs, long-range interactions implicate on average about 60% of all nodes in a PRN. However, it has been proposed that only 30% of residues are crucial for folding (Dill 1999, p.1169).

Proteins exist in 3D physical space. The possibility of a long-range link may depend on some prior sequence of events to bring distant nodes on the polypeptide chain into close spatial proximity with each other. Hence, there is some dependency between links. But long-range links are not mere corollaries to short-range links. Gō (1983) argues that “...folding cannot be a simple unidirectional sequence of events going from smaller to larger structures; long-range interactions also play a determining role in secondary structures and there should be feedback of logic between the levels of organization”.

In our experiments, we observed that a slight delay in the use of long-range links to guide the hill climber significantly improved search performance (Table 4). In delayZ, the use of long-range links is delayed until the fraction of satisfied short-range constraints reaches Z/10. However, if long-range links are included only after all short-range links are satisfied, as in delay10, SR drops to levels similar to SR for onlySE (Table 2), i.e. it is as though long-range links are ignored in the search process completely. These results show that long-range links do help the satisfaction of short-range links in PRN (illustrating Go’s point), but the involvement of long-range

links in the search is more productive once some level of satisfaction (> 50%) in short-range links have occurred.

**Table 4** Result Summary for Experiment 3

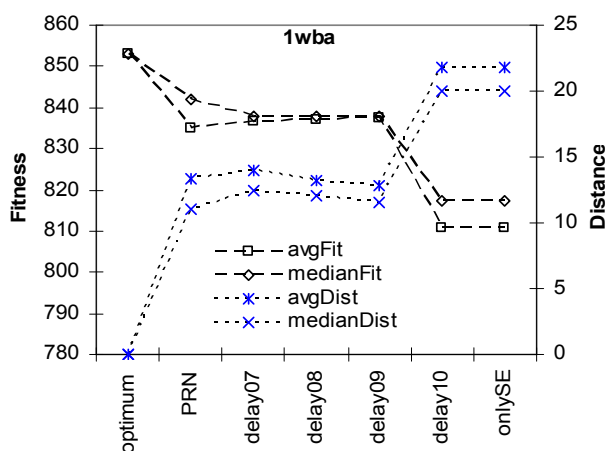
A	B	#	SR	avg_evals
PRN	<b>delay07</b>	54	A < B	A > B
PRN	<b>delay08</b>	54	A < B	A > B
PRN	<b>delay09</b>	53	A < B	A = B

# is the number of networks with SR > 0.0, common to both configurations *A* and *B*. Configurations with better overall search performances are **bolded**. PRNs are from the GH64 dataset. delay07, delay08 and delay09 are, as explained in the text, PRN networks produced by delaying the consideration of long-range links when computing the fitness function until a proportion of short-range links are satisfied. Optimal strings for all networks come from the DSSP output for proteins in the GH64 dataset.

We collected the search points (strings) at the end of failed runs for one PRN (1wba) which failed fairly evenly under the different scenarios tested, and summarized their fitness values and Hamming distances from the global optimum. There is a negative correlation between fitness and distance. We find that runs which did not have enough opportunity to use long-range links to guide the search (onlySE, delay10) produced strings which are most distant and also least fit at the end (Figure 10). Even though the PRN strings are from failed runs, they are still more fit and closer to the global optimum than strings from delay10 or from onlySE.

## Experiment 4

The objective is to observe the effect of varying fitness contribution of links by link length on search performance, given the previous findings. To incorporate the outcome of experiment 3 into the optimization function, we added a weight factor into the *f* term for the experiments in this section



**Figure 10** Summarized fitness and hamming distance from global optimum (optimum) of strings produced by failed runs for 1wba PRN. Number of strings in PRN, delay07, delay08, delay09, delay10 and onlySE are 18, 16, 17, 18, 20 and 20 respectively.

as follows:  $\sum_{i=0}^N g(s_i, \mathbf{s}_i) + \sum_{i=0}^M f(e_i, s, \mathbf{s}, \omega)$ . There are three

options to  $\omega$ : (i) *eq* which assigns equal weight or fitness contribution to all links (this is the option used so far); (ii) *bh* which assigns more weight to links with shorter range; and (iii) *th* which assigns more weight to links with longer range. Let  $e_i$  link nodes  $j$  and  $k$ ,  $d = |j - k|$  and  $|s_j - s_k| = |\mathbf{s}_j - \mathbf{s}_k|$ . If  $\omega = eq$ ,  $f(e_i, s, \mathbf{s}, \omega) = 1$ . If  $\omega = bh$ ,  $f(e_i, s, \mathbf{s}, \omega) = 1/d$ . If  $\omega = th$ ,  $f(e_i, s, \mathbf{s}, \omega) = d$ . However,  $\omega = th$  produced 0.0 SR for GH64 PRNs, and therefore is clearly not a viable fitness function (This outcome is not contradictory to the onlyLE result in Experiment 3 because there, the fitness function assigns equal weights to all links). As such we restrict the following discussion to *bh* and *eq* options.

Compared with the *eq* option (Table 2), the *bh* option significantly improved search performance when PRN is used as the underlying network (Table 6). For both GH64 and EVA132, the number of networks with SR > 0.5 increased by at least 2.5 times (36/14 and 77/27), and there is also an increase of at least 55% in the median size of networks with SR > 0.5. Putting more weight on short-range than long-range links introduces a bias towards the satisfaction of short-range links and is akin to putting a delay on long-range links as we did in the delayZ runs where  $Z < 1.0$  (Experiment 3). However, with the *bh* option, both sets of interactions are present right from the start, so they have more interplay opportunities. And from the results just discussed, there appears to be a payoff to this. Also, PRN<sub>bh</sub> produced significantly better search performance than delay07<sub>eq</sub> (Table 5). Hence, the fitness function with the *bh* option appears to be more compatible to the suitability of PRNs to protein folding.

When  $\omega = bh$ , as in Experiment 1, search performance is still better when the underlying network topology is PRN than when it is Lattice8 (Table 5). However, unlike Experiment 2, search performance is no longer significantly better with randLE than with PRN. For both GH64 and EVA132, when  $\omega$

$= bh$ , there is no significant difference between PRN and randLE in terms of search performance (Table 5). Hence, the shorter APLs that randLE can produce (Figure 8) do not confer a search advantage. With this finding, we observe as in Experiment 1, that shorter APLs do not necessarily guarantee better search performance. However, in this case, no additional links are involved.

**Table 5** Result Summary for Experiment 4

A	B	#	SR	avg_evals
PRN <sub>bh</sub>	PRN	56	A > B	A < B
PRN* <sub>bh</sub>	PRN*	116	A > B	A < B
PRN <sub>bh</sub>	delay07	58	A > B	A < B
PRN <sub>bh</sub>	Lattice8 <sub>bh</sub>	56	A > B	A < B
randLE <sub>bh</sub>	Lattice8 <sub>bh</sub>	56	A > B	A < B
PRN <sub>bh</sub>	randLE <sub>bh</sub>	61	A = B	A = B
PRN* <sub>bh</sub>	randLE* <sub>bh</sub>	122	A = B	A = B

The ‘*bh*’ suffix is used to mark configurations which use the *bh* option; otherwise the default *eq* option is used. PRNs are from the GH64 and where indicated by \*, the EVA132 dataset. Optimal strings for all networks come from the DSSP output for GH64 proteins and where indicated by \*, the DSSP output for EVA132 proteins.

## Summary and Conclusion

In experiment 1, we observed that PRN is a more favourable network topology than Lattice8 for protein folding, but that shorter average path lengths (APLs) need not imply better search performance. In experiment 2, we reported that randomizing long-range links of protein residue networks (randLE) significantly improved search performance over (non-randomized) PRNs. In experiment 3, we found that long-range links play an important role to global optimization and that adding a delay to the involvement of long-range links in the search improved search performance. In experiment 4, we use a modified fitness function which assigns more fitness contribution to shorter links and found that indeed *PRN is a more favourable network topology than randLE for protein folding even though PRN has a significantly longer APL than randLE*.

Shorter APLs in PRNs imply more compactness in native state proteins. That PRNs do not have minimal or at least shorter APLs than they do agrees with the notion that native state proteins are not in the most compact conformation possible (Dill et al, 1995 p. 568).

With  $\omega = bh$  in Experiment 4, *PRNs appear to occupy a sweet spot between complete order and total randomness* PRNs outperformed Lattice8 in terms of search performance, and random graphs (e.g. randSE and randAll networks) are unlikely to produce viable protein structures (Vendruscolo et al 1999). What about randLE which produced a comparable search performance to PRN? randLE represent less plausible 3D structures than PRNs, but more plausible than randSE or randAll (Table 7).

Randomization of long-range links in PRNs (randLE) was performed while keeping node degree of PRNs invariant. Hence, our experiments also suggest that there can be explanations, other than the popular excluded volume argument, beneath the topological limits of PRNs. For



**Table 6** Key summary statistics for results obtained in Experiment 4

PRNs	Configuration	Median, Mean, Sd Success Rate (SR)	Proportion of networks with > 0.0 SR	Median avg_evals of networks with > 0.0 SR	Proportion of networks with > 0.5 SR	Median size of networks with > 0.5 SR
GH64	Lattice8_bh	0.5000, 0.5023, 0.3281	56/64 = 0.8750	4624	31/64 = 0.4844	203
	PRN_bh	0.5750, 0.5750, 0.3176	62/64 = 0.9688	4947	36/64 = 0.5625	208
	randLE_bh	0.5750, 0.5648, 0.3172	62/64 = 0.9688	5336	34/64 = 0.5313	208
	randSE_bh	1.0000, 0.8773, 0.2047	64/64 = 1.0000	3283	57/64 = 0.8906	213
	randAll_bh	1.0000, 0.8352, 0.2426	64/64 = 1.0000	3344	54/64 = 0.8438	212
EVA132	PRN*_bh	0.6000, 0.5614, 0.3176	124/132 = 0.9394	9219	77/132 = 0.5833	226
	randLE*_bh	0.6000, 0.5492, 0.3197	122/132 = 0.9242	9373	76/132 = 0.5758	274

The first column gives the source of the PRN and optimal string *s*. The ‘\*’ indicates the use of the EVA132 dataset. The ‘\_bh’ suffix indicates the use of the *bh* option in the fitness function.

**Table 7** FT-COMAR results for five random PRNs in GH64

PID	N	Lat8	PRN	randLE	randSE	randAll
153L	185	0	0	196	1501	1543
1arb	263	0	0	1319	2278	2221
1cjl	307	0	144	1132	2802	2870
1rpx	690	0	355	3208	6496	6589
1psd	808	0	651	4975	7559	7772

FT-COMAR (Vassura *et al.*, 2008) predicts a plausible 3D construction of a given contact map and threshold, and reports the Hamming distance between the given contact map and the contact map of the predicted structure. Hence, a larger value in this table implies that the given contact map is less plausible as a 3D structure. FT-COMAR works better for thresholds larger than the one we use, i.e. 7 Angstrom, and this affects the results for larger PRNs. Nonetheless, the results still favor PRN over randLE.

instance, both local (high clustering and positive assortativity) and global (short average path length) characteristics of PRN seem necessary to create favorable conditions for protein folding.

Finally, it could be worthwhile, to both protein folding studies and systems organization in general, to understand how the short-range and long-range links in proteins cooperate to create mutual satisfaction without either set necessarily dominating the process.

## References

- Anfinsen, C.B. (1973) Principles that Govern the Folding of Protein Chains. *Science* 181, pp. 223-230.
- Atilgan, A.R., Akan, P., and Baysal, C. (2004) Small-world Communication of Residues and Significance for Protein Dynamics. *Biophysics Journal* 86, pp. 85-91.
- Bagler, G., and Sinha, S. (2007) Assortative Mixing in Protein Contact Networks and Protein Folding Kinetics. *Structural Bioinformatics* 23(14), pp. 1760-1767.
- Bartoli, L., Fariselli, P., and Casadio, R. (2007) The Effect of Backbone on the Small-world Properties of Protein Contact Networks. *Physical Biology* 4, pp. L1-L5.
- Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., and Bourne, P.E. (2000) The Protein Data Bank. *Nucleic Acids Research* 28, pp. 235-242. <http://www.rcsb.org/pdb>
- Bryngelson, J.D., and Wolynes, P.G. (1987) Spin Glasses and the Statistical Mechanics of Protein Folding. *PNAS USA* 84, pp. 7524-7528
- Del Sol, A., Fujihashi, H., Amoros, D., and Nussinov, R. (2006) Residues Crucial for Maintaining Short Paths in Network Communication Mediate Signaling in Proteins. *Molecular Systems Biology* 2.
- Dill, K.A. (1999) Polymer principles and protein folding. *Protein Science* 8, pp. 1166-1180.
- Dill, K.A., Bromberg, S., Yue, K., Fiebig, K.M., Yee, D.P., Thomas, P.D. and Chan, H.S. (1995) Principles of protein folding – A perspective from simple exact models. *Protein Science* 4, pp. 561-602
- Dokholyan, N.V., Li, L., Ding, F., and Shakhnovich, E.I. (2002) Topological Determinants of Protein Folding. *PNAS* 99(13), pp. 8637-8641.
- Gaci, O., and Balev, S. (2009) Node Degree Distribution in Amino Acid Interaction Networks. In: *Proc. Computational Structural Bioinformatics Workshop*, Washington D.C. USA.
- Gö, N. (1983) Theoretical Studies of Protein Folding. *Ann. Rev. Biophys. Bioeng.* 12, pp. 183-210
- Greene, L.H., and Higman, V.A. (2003) Uncovering Network Systems within Protein Structures. *Journal of Molecular Biology* 334, pp. 781-791.
- Kabsch, W., and Sander, C. (1983) Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-bonded and Geometrical Features. *Biopolymers* 22(12), pp. 2577-2637. <http://swift.cmbi.ru.nl/gv/dssp>
- Karplus, M. (1997) The Levinthal Paradox: Yesterday and Today. *Folding & Design* 2, pp. S69-S75.
- Levinthal, C. (1969) How to Fold Graciously. In: Debrunner, J.T.P., and Munck, E. (eds.) *Mossbauer Spectroscopy in Biological Systems: Proceedings of a Meeting held at Allerton House*. Monticello, Illinois. University of Illinois Press, pp. 22-24.
- Maslov, S., and Sneppen, K. (2002) Specificity and Stability in Topology of Protein Networks. *Science* 296 pp. 910-913.
- Newman, M.E.J. (2002) Assortative Mixing in Networks. *Physical Review Letters* 89, 208701.
- Rost, B. (1999) Twilight Zone of Protein Sequence Alignments. *Protein Engineering* 12(2), pp. 85-94. [http://cubic.bioc.columbia.edu/ava/res/unique\\_list.html](http://cubic.bioc.columbia.edu/ava/res/unique_list.html) (Sep 23 2010).
- Vassura, M., Margara, L., Di-Lena, P., Medri, F., Farisello, P., and Casadio, R. (2008) Reconstruction of 3D Structures from Protein Contact Maps. *IEEE/ACM Trans. On Comp. Biol. And Bioinfo.* 5(3) pp. 357-367. <http://bioinformatics.cs.unibo.it/FT-COMAR>
- Vendruscolo, M., Subramanian, B., Kanter, I., Domany, E., and Lebowitz, J. (1999) Statistical Properties of Contact Maps. *Physical Review E* 59, 977-84.
- Vendruscolo, M., Dokholyan, N.V., Paci, E., and Karplus, M. (2002) Small-world View of the Amino Acids that Play a Key Role in Protein Folding. *Physical Review E* 65, 061910
- Watts, D.J., and Strogatz, S.H. (1998) Collective Dynamics of ‘small-world’ Networks. *Nature* 393, pp. 440-442.
- Watts, D.J. (1999) *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton University Press, Princeton, NJ.
- Zwanzig, R., Szabo A., and Bagchi, B. (1992) Levinthal’s Paradox. *Proc. Natl. Acad. Sci. USA* 89, pp. 20-22.

# A computational model of ant nest morphogenesis

Anaïs Khuong<sup>1,2,\*</sup>, Guy Theraulaz<sup>1,2</sup>, Christian Jost<sup>1,2</sup>, Andrea Perna<sup>1,2,3</sup>, Jacques Gautrais<sup>1,2</sup>

<sup>1</sup> Centre de Recherches sur la Cognition Animale, UMR-CNRS 5169, Université Paul Sabatier, Bat 4R3, 118 Route de Narbonne, 31062 Toulouse cedex 9, France.

<sup>2</sup> CNRS, Centre de Recherches sur la Cognition Animale, F-31062 Toulouse, France.

<sup>3</sup> Collective Behaviour Group, Matematiska Institutionen, Uppsala Universitet, Box 480 - 75106 Uppsala, Sweden.

\* corresponding author: akhuong@cict.fr

## Abstract

The nests of social insects are not only impressive because of their sheer complexity, but also because they are built from much smaller agents whose work is not centrally coordinated. A central question is therefore how this coordination can lead to such large scale structures. In this paper we present an individual based nest construction model from experimentally inspired rules. The coordination of the building process is achieved through three main ingredients: 1) stigmergy, which implies that the local configuration of the structure is the stimulus which determines how to continue, 2) body template, where the interaction between the ant's body and the growing structure determines the proportions of the emerging pattern, and 3) a construction "pheromone", a chemical compound capable of triggering building actions. Our simulations show that this simple set of coordination rules can reproduce the key features observed experimentally in the ant *Lasius niger*, notably the emergence of mushroom-like pillars and layered structures. A sensitivity analysis on the evaporation rate of the construction pheromone shows that a large range of architectures, from dynamic multilayered nests to compact sponge-like structures, can be produced with the same behavioural rules by simply modifying evaporation rate. We discuss the relevance of these results with respect to the variety of nest architectures found in social insects.

## Introduction

The nest architectures of social insects (ants, termites, some bees and wasps species) are among the most impressive and complex artifacts built by animals with the notable exception of man (Theraulaz et al., 1998; Turner, 2000a,b). All along the evolution of social insects, there has been a whole set of innovations in terms of architectural designs and construction techniques that proved to be very efficient to solve a large number of problems such as controlling the temperature inside the nest, ensuring the gas exchanges with the outside environment (Bollazzi and Roces, 2007) or adapting the nest structure when colony size is growing (Hansell, 2005). More than fifty years after Pierre-Paul Grassé has introduced stigmergy as a basic principle for the coordination of work in these societies, we are still very far from having a full understanding of the mechanisms that shape architecture and functional designs of the nests (Grassé, 1959). While being

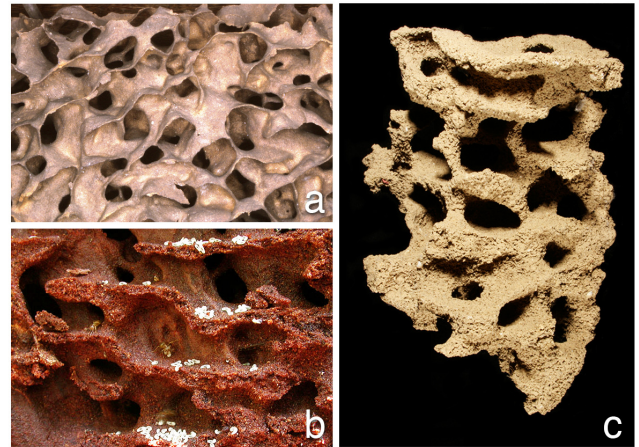


Figure 1: Examples of nest architectures built by ant colonies. (a) Detail of a nest in wood pulp sculpted by the ant *Lasius fuliginosus*. (b) Detail of the nest structure built by the ant *Lasius pallitarsis* ©Alex Wild. (c) A closer look on the walls and vertical passages connecting chambers inside a nest built by the ant *Lasius niger*.

extremely simple, stigmergy is able to give rise to complex self-organised patterns (Deneubourg, 1977; Bonabeau et al., 1998). Moreover stigmergy is often combined with environmental templates that modulate the expression of individual building rules, thus increasing the range of potential architectures (Jost et al., 2007). Other factors are also likely to play a key role in nest morphogenesis such as building pheromones. Such pheromones have been identified in termites (Bruinsma, 1979) and are likely to exist in some ant species (our unpublished data).

In this paper we present an individual-based model of ant nest construction based on a detailed analysis of individual building behaviours that take into account the logistic constraints imposed by the architecture on the movement of ants. With this model we investigate the role played by the building pheromone on the resulting shape of the nest.

The paper is organised as follows: in section two we introduce the experimental results and describe the individual building rules in the ant *Lasius niger*. In section 3, we present an overview of the 3D agent-based model. In section 4, we report simulation results that illustrate the impact of evaporation rate of building pheromone on the resulting nest architectures. Finally, in section 5, we establish comparisons with related work and draw some conclusions and directions for future work.

## Construction mechanisms at the individual level

We performed a series of experiments to investigate the mechanisms involved in nest construction in the ant *Lasius niger*. These experiments showed that the deposition of material in a particular place stimulates the ants to accumulate more material in that same place, thus creating a positive feed-back. Experiments also revealed that the workers add a chemical signal (a building pheromone) to the building material. The main action of this chemical signal is to attract ants, but there are also indications that it stimulates the deposition of building material.

There was no particular effect of this signal on the extraction of building material, but it was noticed that ants prefer to dig where they have already dug, forming a type of quarry. This may be simply due to physical constraints, in the sense that it is much harder for an ant to extract a pellet from a place where the soil has been solidly packed, compared to a place where the soil surface has been broken (as by a previously digging ant).

The consequence of all these behaviours is the formation of pillars. Once pillars have been erected and have reached a critical height, the workers start to build a canopy on the sides. The height at which the ants attach the pellets on the pillars corresponds approximately to the mean body length of an ant worker. The workers therefore use their body as a kind of template to decide at which height they will stop to increase the size of an existing pillar and start to build a roof from that pillar.

## Behaviour-based model of nest construction

We developed a spatially explicit individual-based model in a discrete 3D cubic-lattice in which we have incorporated the behavioral rules characterized by the experiments

### General principles

The model is stochastic: ant workers are represented by agents whose behavioural rules are modelled according to probabilities to perform simple elementary actions. Moreover, the process is Markovian: the probability of performing a given action is only depending on the current state of the environment around the agent (spatial configuration, quantity and age of the building pheromone, number of

empty cells below). Indeed, agents are memoryless and tireless.

Following Ladley and Bullock's work (2005), our model takes into account the geometric constraints: each pellet of building material occupies a single cell and the ants are represented by agents that move randomly in a three-dimensional discrete cubic lattice ( $200 \times 200 \times 200$  voxels). Each agent occupies a single cell and their movements are constrained by the structures they build: they cannot walk through the built structures. The layers on the bottom and on the sides of the lattice act as borders. Ants simply bounce on the floor and walls when they come into contact. We choose a discrete time step approach. At each step, the system is updated: agents move, then, if they are not already transporting building material, they can pick up a pellet, else drop it, or simply continue their walk without doing anything else. Each agent can only perceive the first twenty six neighbouring cells that surround the place where it is located at a given moment (cell  $c$ ). We denote these twenty-six 3D-neighbours by  $V_{26,c}$ , and the influent neighbourhood for certain behavioural rules may be restricted as detailed below.

### Behavioural rules of an ant

**Motion** The motion of ants is a constrained random walk, which means that they stay in contact with the outer surface of the architecture. The building pheromone that will be introduced in dropped pellets doesn't affect their motion: ants are not attracted or repelled by it.

Ants may only move to adjacent locations, i.e. the six orthogonal cells. We call  $V_{6,c}$  this reduced neighbourhood around the cube  $c$ . A worker cannot walk through an occupied cell (clay, floor, wall or another worker): only empty cells of  $V_{6,c}$  are really available for moving. The second constraint prevents flying ants: they must stay in contact with the surface of the structure. Thus, only adjacent locations, which have at least one  $V_{6,c}$  neighbour cell occupied by clay, floor or walls, are available for moving. The algorithmic description of the motion rule is summarized below (Algorithm 1).

**Picking-up rule** A worker can only pick up a pellet when it stands atop it. If it does, it takes the location of the pellet.

To compute the picking-up probability, the worker simply considers the bottom layer of cells in its neighbourhood. We call  $V_{8,c}$  the eight horizontal neighbours of the cell she is standing on. The probability to pick-up the block she is standing on is not influenced by the presence of pheromone in the material but it slightly decreases as the quantity of building material in this bottom layer increases. This is a simple consequence of the fact that it is much more difficult for an ant to extract a pellet when the ground is packed more solidly. The corresponding picking up probability is shown in Figure 2 (a).

**Algorithm 1** Motion rule – The algorithm used to simulate the workers’ random walk. To simulate agents’ diffusion, one agent performs *nbMove* elementary moves each time step.  $c_w$  is the cube with the worker  $w$ .  $V_{6,w}$  is the list of the six neighbours of  $w$  that share a face with it. We denote by  $A_w$  the list of the accessible immediate neighbours of  $w$  and by Random the random drawing in a discrete set of cells.

```

1: // The worker is in  $c_w$ .
2: for all step  $\in \{1; nbMove\}$  do
3:    $A_w = \text{EMPTYLIST}$ 
4:   for all  $c_i \in V_{6,w}$  do
5:     if ( $c_i == \text{empty}$ ) and (one of  $V_{6,c_i}$  is full)
        // The cube  $c_i$  is accessible.
        then
6:        $A_w \leftarrow \text{concat}(A_w, c_i)$ 
7:     end if
8:   end for
        //  $A_w$  contains all the accessible adjacent neighbours
        // of  $w$ .
        // Random choice of  $c_r \in A_w$ .
9:    $c_r = \text{Random}(A_w)$ 
10:   $c_w \leftarrow c_r$ 
11: end for

```

The algorithmic description of the picking up rule is summarized below (Algorithm 2).

**Building rule** A worker drops its pellet at its current location, provided there exists a cell in the neighbourhood  $V_{26,c}$  where it can move after dropping. This building behaviour is also conditioned by physical constraints, which means that a building pellet can be added to the previous structure only if at least one of its faces is in contact with another pellet located in the neighbourhood (pellets do not stick together by the cube’s corners or edges).

Since we found experimental evidence that clay which has been previously manipulated by workers stimulates the dropping behaviour, we implement a building pheromone. The building pheromone contained in a pellet is renewed each time a pellet has been picked-up and dropped. It does not diffuse to adjacent cells but still undergoes an exponential decay (at some rate), so it directly delivers a local signal about the time elapsed since the pellet was deposited. The probability to drop a pellet or add it to an existing structure is enhanced by the number of pellets previously dropped in the neighbourhood but it decreases with time.

Since we found also experimental evidence that ants use their body as a kind of template to build the canopies on top of the pillars and prefer to drop their load at some height (standing upright along the pillar), we also include a modulation of the dropping probability when a worker is moving over a vertical surface. In those situations, when the potential dropping site has  $h$  empty cells below it, the behavioural

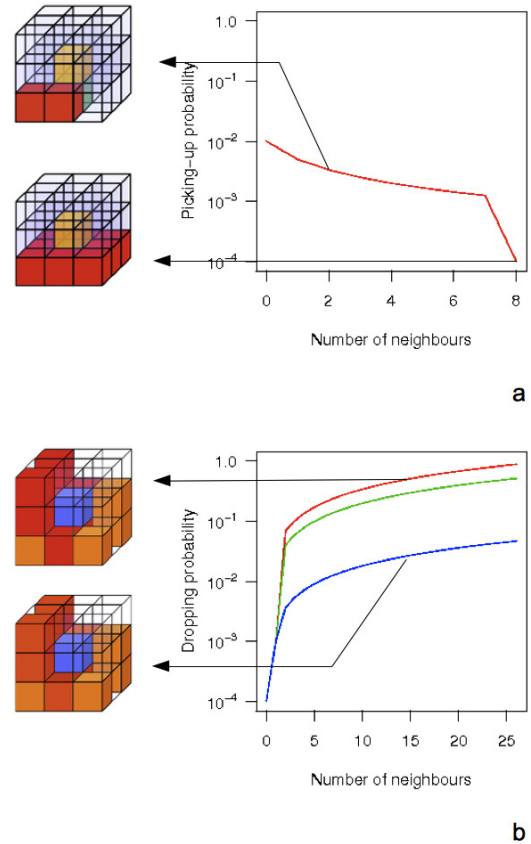


Figure 2: Probabilistic building rules implemented in the model. (a) Picking-up probability as a function of the number of cells containing clay in the bottom layer. The shown curve is implemented as a two-parameter function of the number of neighbours  $n$ , taking the value *spontPick* for  $n = 0$ , *spontPick*/100 for  $n = 8$  and *spontPick*/(*amplifPick* ·  $n$ ) for  $1 \leq n \leq 7$ . (b) Dropping probability as a function of the local density (number of neighbouring cells containing clay,  $n$ ) and of the age of the latest dropped pellet in the neighbourhood. It takes the values *spontDrop* for  $n = 0$  and  $(\text{drop1} + \text{amplifDrop} \cdot (n - 1)) \cdot \exp(-(time - \text{latestDropTime}) \cdot \text{evap})$  for  $1 \leq n \leq 26$ , where *time* is the current time. For parameter values and explanations see Table 1. The red, green and blue lines mark the probabilities where the last dropped pellet is younger to older respectively.

**Algorithm 2** Picking-up rule – The algorithm used to estimate the granularity around the worker  $w$  currently located in the cube  $c_w$ . We denote by  $t_{w,p}$  the target cube for the picking up. Here, the rule  $H_{p,1}$  is that  $t_{w,p}$  is underneath  $c_w$ .  $V_{8,t_{w,p}}$  is the list of the influent neighbours of  $t_{w,p}$ . The variable  $n_{w,p}$  counts the number of full neighbours.  $p_{w,n}$  is the associated picking up probability. Uniform denotes a random number in  $[0; 1[$ .

---

```

1:  $n_{w,p} = 0$ 
2: for all  $c_i \in V_{8,t_{w,p}}$  do
3:   if  $c_i == \text{full}$  then
4:      $n_{w,p} \leftarrow n_{w,p} + 1$ 
5:   end if
6: end for
  // Calculate the picking up probability, associated to
   $n_{w,p}$ .
7:  $p_{w,p} \leftarrow f(n_{w,p})$  // According to a decreasing amplifi-
  cation curve, see figure 2 (a).
8: if  $p_{w,p} < \text{Uniform}$  then
9:   // Pick-up  $t_{w,p}$ .
10:  // Move to  $t_{w,p}$ .
11: end if

```

---

**Algorithm 3** Building rule – The algorithm used to estimate the local density around the worker  $w$  which is located in the cube  $c_w$ . The influent neighbours are  $V_{26,c_w}$ . The variable  $n_{w,d}$  allows to count the number of full neighbours.  $ageOf(c_i)$  corresponds to the date of dropping the pellet  $c_i$ . We define by  $latestDropAge$  the date of the latest dropped pellet in  $V_{26,c_w}$ . Uniform denotes a random number in  $[0; 1[$ .

---

```

1:  $n_{w,d} = 0$ 
2:  $latestDropAge = 0$ 
3: for all  $c_i \in V_{26,c_w}$  do
4:   if  $c_i == \text{full}$  then
5:      $n_{w,d} \leftarrow n_{w,d} + 1$ 
6:     if  $latestDropAge < ageOf(c_i)$  then
7:        $latestDropAge = ageOf(c_i)$ 
8:     end if
9:   end if
10: end for
  // Calculate the dropping probability, associated to  $n_{w,d}$ 
  and  $latestDropAge$ .
11:  $p_{w,d} \leftarrow f(n_{w,d}, latestDropAge)$  // According to the
  increasing curve shown in figure 2 (b).
12: if  $p_{w,d} < \text{Uniform}$  then
13:   // Drop in  $c_w$ .
14:    $ageOf(c_w) = \text{currentStep}$ 
15:   // Use OneMove (Algorithm 1) with  $nbMove = 1$  to
  escape from  $c_w$ .
16: end if

```

---

algorithm includes a multiplication factor of the dropping probability,  $p_{w,d}(h)$ , according to the equation

$$p_{w,d}(h) = p_{w,d} \cdot \frac{h^n}{\bar{h}^n + h^n} \quad (1)$$

with  $\bar{h}$  being the mean length of an ant.

The algorithmic description of the building rule is summarised above (Algorithm 3).

## Simulation results

We implemented the three behavioural rules in the model and we run the simulations with the parameters values given in table 1. The 3D cubic lattice ( $200 \times 200 \times 200$ ) was initialised with 20 bottom layers uniformly filled with pellet and 1000 workers randomly placed on the surface. The maximum value of the spontaneous picking-up probability is reached when the eight horizontal neighbours on the bottom layer are empty. This is fixed in the picking-up probability function by the parameter *spontPick* which we set to  $10^{-2}$ . The decrease in picking-up rate, specified by parameter *amplifPick*, is set to 1.

The spontaneous dropping probability *spontDrop*, when there is only one pellet in its  $V_{26,c}$  neighbourhood, is fixed at  $10^{-4}$ . In case of one additional neighbouring cell we set the dropping probability to *drop1* =  $10^{-3}$  (Fig 2). Dropping probability then increases continuously with the number of pellets in  $V_{26,c}$  to the maximum value of *drop26* = 0.9. The evaporation rate is initially set to *evap* =  $1.6 \times 10^{-5}$  per time step and then modified to explore its impact on the emerging 3D architectures.

Model parameter	Description	Value
<i>spontDrop</i>	Spontaneous dropping probability	$10^{-4}$
<i>drop1</i>	Dropping probability in the case of exactly one marked neighbour	$10^{-3}$
<i>amplifDrop</i>	Factor modulating the dropping probability	0.036
<i>evap</i>	Evaporation rate of the building pheromone	$3.2 \times 10^{-4}$ to $8.0 \times 10^{-7}$
<i>spontPick</i>	Spontaneous picking-up probability	$10^{-2}$
<i>amplifPick</i>	Factor modulating the picking-up probability	1.0

Table 1: Parameters values used in simulations. Rates and probabilities apply to one time step. See Fig 2 for the use of these parameters.

## Pillars and roofs

When running this model one can observe the formation of pillars. When the height of the pillars becomes high enough, pellets are added on the pillars' sides; this rapidly increases



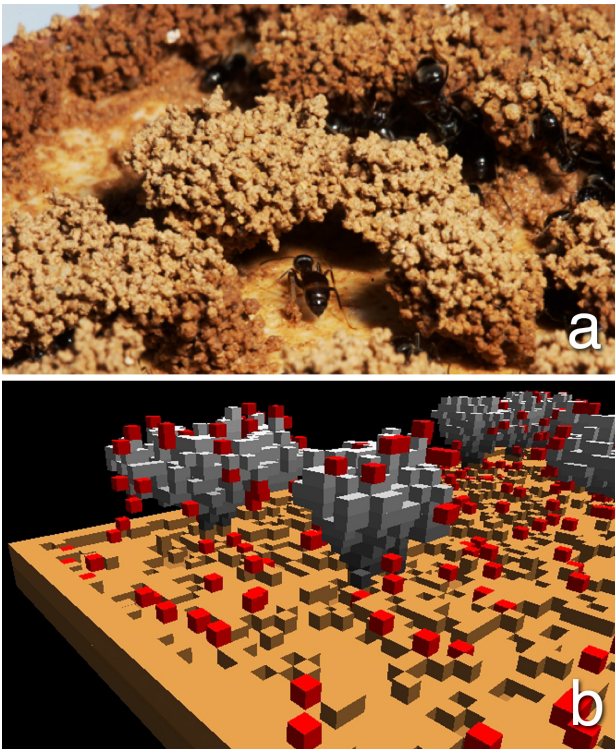


Figure 3: A comparison of the structures built in experiments and in a simulation of the model. (a) An arch that covers a passage between two pillars built by ants in experimental conditions. (b) A close view of a simulation result with the same initial conditions as the ones used in experiments. The model is able to reproduce the characteristic shapes observed in these experiments. The parameter values used in this simulation are listed in table 1.

the surface of the pillar top, leading to the formation of some kind of hat or roof. These roofs look quite similar to the ones we got in the experiments (see figure 3). Moreover, when two roofs are close enough to each other, they can merge. The result is an arch that covers a passage between the two pillars.

Physical stress due to gravity and decreasing cohesion due to evaporating water should finally lead to collapsing roofs and pillars, an event that was sometimes observed in the experiments. The current version of the model does not include these processes, but when inter-pillar distance is not too large this should not be much of a problem.

### Effect of the decay of the building pheromone on 3D architectures

To explore the diversity of nest architectures the model is able to produce, we investigate the role of the decay rate associated with the building pheromone. This is already known to be a key ingredient of the self-organization in social insects, it has a major impact on the collective dy-

namics such as: trail formation and path choice in ants (Goss et al., 1989; Beckers et al., 1992; Jeanson et al., 2003; Sumpter and Beekman, 2003), construction of pillars in termites (Deneubourg, 1977; Franks and Deneubourg, 1997; Bonabeau et al., 1998), construction of wall in ants (Franks et al., 1992), digging networks of galleries in ants (Buhl et al., 2005).

Figure 4 shows that the evaporation rate of the building pheromone is indeed a highly influential parameter on the resulting structures.

When there is a strong evaporation rate ( $evap = 3.2 \times 10^{-4}$ ), the final structure is laminar (figure 4 (a)). In the early steps, agents begin building several tiny pillars on the unmarked initial surface. They cover them with thin roofs or capitals. The surface of these roofs increases, several roofs merge, forming a thin first layer that becomes the first floor. In the next steps, the construction dynamics undergo the same cycle of events, leading to a new floor. Moreover agents can enlarge the pillars by adding new pellets on their sides.

When the evaporation rate is smaller ( $evap = 1.6 \times 10^{-5}$ ), the structure is still laminar, but the layers are less plane than in the previous case (figure 4 (b)). The initial phase is similar to the previous condition, but there is a larger number of pillars and the capitals are thicker. After the completion of the first floor, the construction of new pillars occurs at a faster rate than with the higher evaporation rate. A closer look at the growth and the evolution of the nest structure reveals that while the whole structure remains quite similar in time, it is constantly destroyed and rebuilt. The consequence of this remodelling process, in which the ants destroy what they have built previously, is a progressive drift of all the layers from the top to the bottom. It seems that a kind of wave runs through the whole structure. These traveling waves are indeed the simple consequence of the fact that the only place where the ants can pick up some building material is the bottom layer. So it quite naturally induces a kind of symmetry breaking in the remodelling process.

Finally when the evaporation rate is very low ( $evap = 8 \times 10^{-7}$ , figure 4 (c)), the model leads to a sponge-like structure that looks similar to the nest built by *Lasius niger*. In a first step, pillars also emerge and are covered with capitals that are more spherical than in the two previous cases. Thus, when the capitals merge, the layer is thicker. In a second step, pellets can be dropped anywhere on the new floor. No pillar emerges in this case, the layer is just thickened. Sometimes, a little heap appears by chance, a new pillar is built and starts to grow. This new pillar merges quickly with the structure in its vicinity, leading to the formation of a chamber. The next floor is built when many chambers have been created and closed. In the next steps, the construction dynamics undergoes the same cycle of events: (1) thicken the floor ; (2) emergence of few little pillars ; (3) fusion of the roofs, which leads to the formation of chambers.

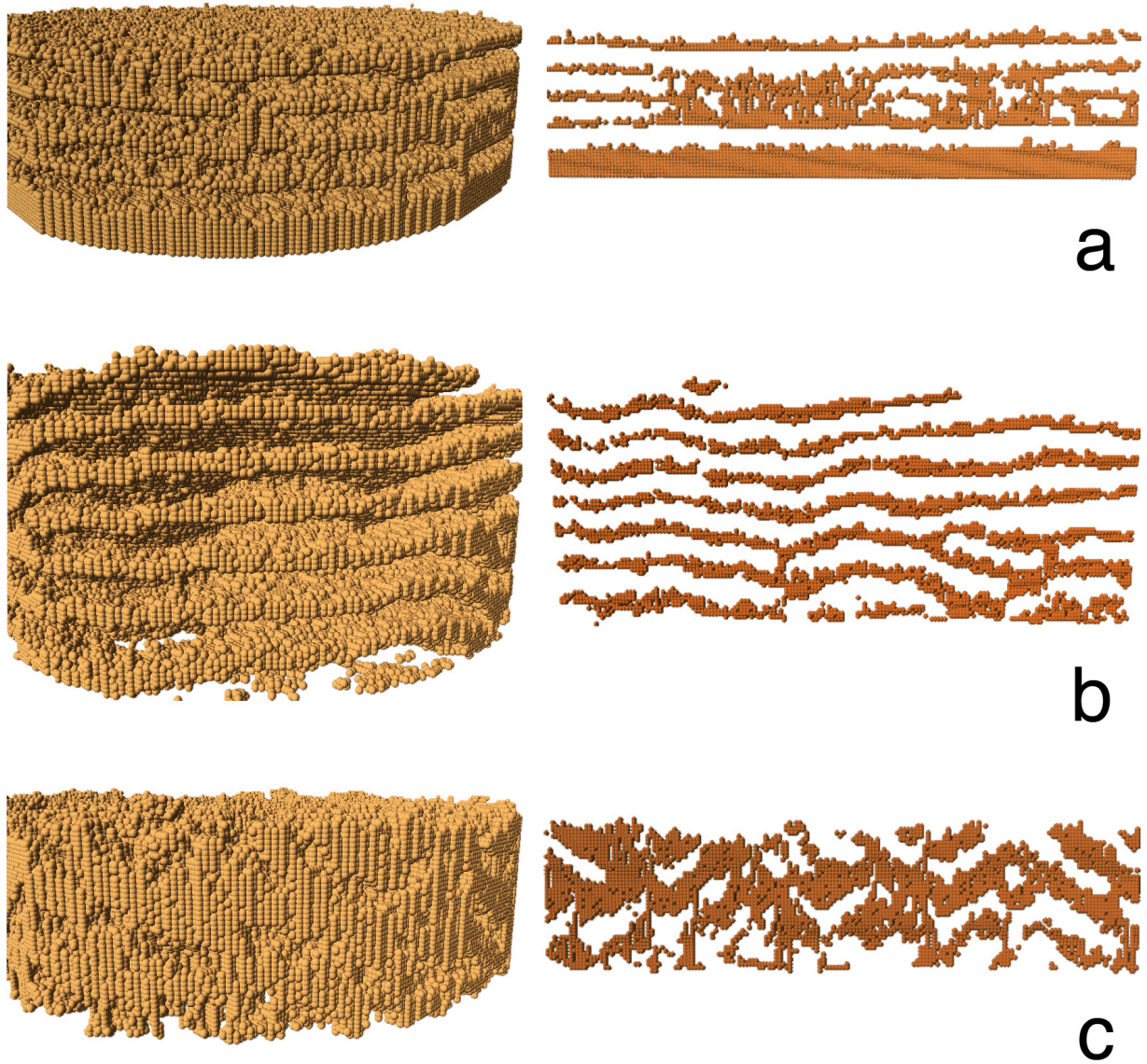


Figure 4: The influence of the evaporation rate of the building pheromone on the nests structure. Left : 3D structure. Right : Vertical cut ( $x \in [98; 101]$ ). (a) With a strong evaporation rate ( $evap = 3.2 \times 10^{-4}$ ), the construction process leads to the formation of a laminar structure. The horizontal layers are connected with thick pillars. (b) With an intermediate evaporation rate ( $evap = 1.6 \times 10^{-5}$ ), the structure is still laminar, but sometimes two successive layers can intersect and form a ramp that connects successive floors (c) When the evaporation rate is very low ( $evap = 8 \times 10^{-7}$ ) we get a sponge-like structure.



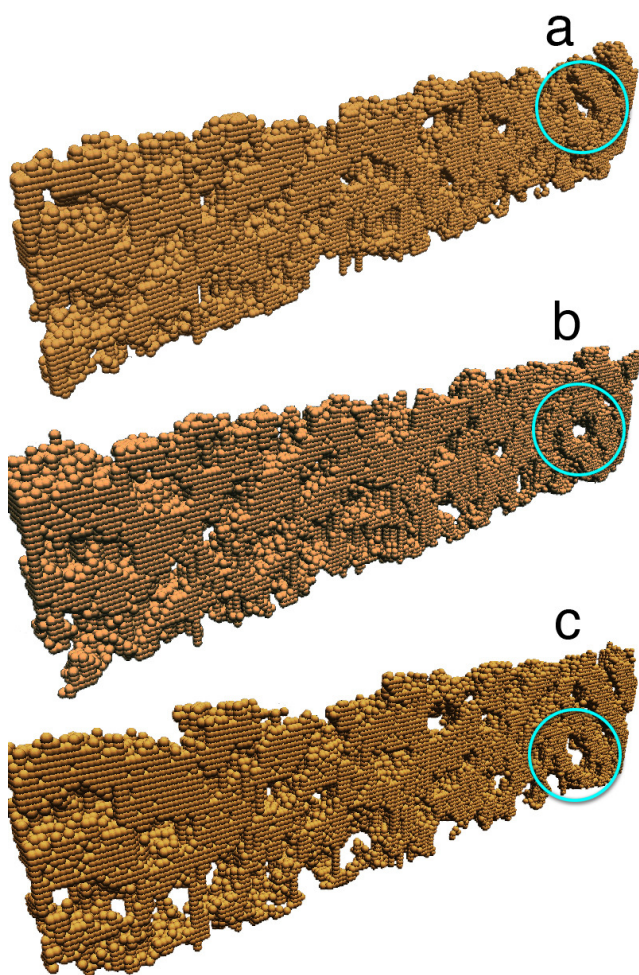


Figure 5: A closer look on the building dynamics in the case of a very weak evaporation rate ( $8 \times 10^{-7}$ ). (a) to (c) show three successive steps of a simulation. The circle marks a chamber "moving" downwards.

## Discussion and conclusion

In this paper we introduced a 3D model of collective ant nest construction. This model is based on stochastic individual rules derived from the experimental analysis of building behaviour in the ant *Lasius niger*. The model also integrates logistic constraints, that is physical limitations on the movement of ants imposed by the nest architecture. Such constraints have been previously implemented by Ladley and Bullock (2005) to simulate the formation of the royal chamber and covered lanes in termites.

There are two main differences with this previous work. First, in our model there is no chemical template created by the diffusion of pheromones. This contrasts with termites, because the queen releases a pheromone that controls the distance at which workers start to build. At the very begin-

ning, this chemical template strongly interacts with the self-organizing building processes. And this combination gives rise to pillar-like structures formed at roughly regular spatial intervals, but at a specific distance from the queens body. In ants, the effect of the body-template begins to work later in the construction process, when the pillar-like structures have reached a critical size. The consequence is the formation of a double regular spatial pattern: the first one characterizes the spatial distribution of the pillars and the second one characterizes the layered structure of the nest. The second difference is a constant remodelling process that results from the ants activity. In our model, ants continuously destroy what they have built previously. Once a layer is in place, all its surface is eroded as a consequence of the ants digging activity and rapidly the material accumulates on the underneath surface. As a main consequence all the layers drift progressively downwards. And the speed of the travelling and remodelling wave results from a balance between the net deposition rates of building material at the upper and lower surfaces of a layer.

Sometimes, ants may accumulate by chance a little bit more material on the underneath surface of an existing layer. This gives rise to a new pillar growing from top to bottom. Once this pillar is built, it remains in place because the virtual ants can only dig on the bottom layer and not on the sides. This creates a kind of defect that propagates within the structure as the remodelling process goes on. The same process also produces connection areas between different layers close to these pillars. The motion of ants is in turn channeled by the spatial distribution of these connection areas. Then, depending on the evaporation rate, this channeling process may also promote the deposition of building material on the edges of the pillars, thus changing their size and shape.

Our model showed that the resulting nest structure strongly depends on the evaporation rate of the building pheromone. When the evaporation rate is very high ( $evap = 3.2 \times 10^{-4}$ ), only the very latest depositions of material can enhance the accumulation of more material. In these conditions, only a small number of pillars can be built, there is a strong competition among pillars to attract builders and the distance between pillars increases. The second consequence is a much more important enlargement of the capitals on top of the remaining pillars. As soon as a capital is built, the material is deposited at a faster rate on its border and the resulting shape of the roof becomes flat and thin. When the evaporation rate is less important ( $evap = 1.6 \times 10^{-5}$ ), the number of pillars increases and the enlargement of the capitals on top of the pillars is also much more important. Each pillar becomes a seed from which a new layer is growing. Since at each level there exist several seeds from which different layers are growing, it may happen that one of these layers collides with another one that is a part of the next level below. This results in the formation of inter-crossings of ceilings and floors belonging to two successive layers, leading

to the formation of ramps. Finally, when there is a very weak evaporation of the building pheromone ( $evap = 8 \times 10^{-7}$ ), the enlargement of the capitals becomes even more important. After having merged the capitals still increase their size and the floor of the new layer is so attractive that the depositions of building material occur more or less uniformly over the whole surface. Instead of well-defined pillars and floors, ants build globular structures enclosing empty and irregular chambers. The whole structure adopts a sponge-like structure.

The same kind of architectural diversity is observed in *Lasius* species. The ant *Lasius fuliginosus* builds a sponge-like nest (Figure 1 (a)), whereas *Lasius pallitarsis* and *Lasius niger* nests show layered structures (Figures 1 (b) and (c)). Our model shows that the same mechanisms can account for significant changes in the nests shape.

These variations may have several origins: it might be a consequence of the variation of environmental conditions (e.g. temperature and humidity levels). If these conditions change, the same species will be able to build nest structures that look very different, for example in *Acromyrmex* ants (Bollazzi et al., 2008) or in *Macrotermes* termites (Korb, 2003). But this variation may also result from the physical properties of the building pheromone itself. In particular one may imagine that different species of ants or termites can use similar building rules but different chemical cues. Physical and chemical properties of the building pheromone could thus play a key role in the diversity of nest architectures built by ants and termites. This is an important issue that needs to be addressed in future experimental work.

## Acknowledgements

Anaïs Khuong was supported by a PhD fellowship from The Cluster for Higher Education and Research "Université de Toulouse" (PRES) and Région Midi-Pyrénées. This study was supported by a grant from the MESOMORPH project (Grant No. ANR-06-BYOS-0008).

## References

- Beckers, R., Deneubourg, J.-L., and Goss, S. (1992). Trail laying behaviour during food recruitment in the ant *Lasius niger* (L.). *Insectes Sociaux*, 39:59–72.
- Bollazzi, M., Kronenbitter, J., and Rocas, F. (2008). Soil temperature, digging behaviour, and the adaptive value of nest depth in south american species of *Acromyrmex* leaf-cutting ants. *Oecologia*, 158:165–175.
- Bollazzi, M. and Rocas, F. (2007). To build or not to build: circulating dry air organizes building responses for climate control in the leaf-cutting ant *Acromyrmex ambiguus*. *Animal Behaviour*, 74:1349–1355.
- Bonabeau, E., Theraulaz, G., Deneubourg, J.-L., Franks, N. R., Rafelsberger, O., Joly, J.-L., and Blanco, S. (1998). A model for the emergence of pillars, walls and royal chambers in termite nests. *Philosophical Transactions of the Royal Society of London*, 353:1561–1576.
- Bruinsma, O. H. (1979). *An Analysis of Building Behaviour of the Termite Macrotermes subhyalinus*. PhD thesis, Lanbouwhogeschool Wageningen, The Netherlands.
- Buhl, J., Deneubourg, J.-L., Grimal, A., and Theraulaz, G. (2005). Self-organized digging activity in ant colonies. *Behavioral Ecology and Sociobiology*, 85:9–17.
- Deneubourg, J.-L. (1977). Application de l'ordre par fluctuations à la description de certaines étapes de la construction du nid chez les termites. *Insectes Sociaux*, 24:117–130.
- Franks, N. and Deneubourg, J.-L. (1997). Self-organizing nest construction in ants: Individual worker behaviour and the nest's dynamics. *Animal Behaviour*, 54:779–796.
- Franks, N., Wilby, A., Silverman, V. W., and Tofts, C. (1992). Self-organizing nest construction in ants: sophisticated building by blind bulldozing. *Animal Behaviour*, 44:357–375.
- Goss, S., Aron, S., Deneubourg, J.-L., and Pasteels, J. M. (1989). Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76:579–581.
- Grassé, P. (1959). La reconstruction du nid et les coordinations inter-individuelles chez *Bellicositermes Natalensis* et *Cubitermes* sp. la théorie de la stigmergie : Essai d'interprétation du comportement des termites constructeurs. *Insectes sociaux*, 6:41–81.
- Hansell, M. (2005). *Animal Architecture*. Oxford University Press, New-York.
- Jeanson, R., Ratnieks, F. L. W., and Deneubourg, J.-L. (2003). Pheromone trail decay on different substrates in the pharaoh's ant, *Monomorium pharaonis*. *Physiological Entomology*, 28:192–198.
- Jost, C., Verret, J., Casellas, E., Gautrais, J., Challet, M., Lluc, J., Blanco, S., Clifton, M., and Theraulaz, G. (2007). The interplay between a self-organized process and an environmental template: corpse clustering under the influence of air currents in ants. *Journal of the Royal Society Interface*, 4:107–116.
- Korb, J. (2003). Thermoregulation and ventilation of termite mounds. *Naturwissenschaften*, 90:212–219.
- Ladley, D. and Bullock, S. (2005). The role of logistic constraints in termite construction of chambers and tunnels. *Journal of Theoretical Biology*, 234:551–564.
- Sumpter, D. J. and Beekman, M. (2003). From nonlinearity to optimality: pheromone trail foraging by ants. *Animal behaviour*, 66:273–280.
- Theraulaz, G., Bonabeau, E., and Deneubourg, J. (1998). The origin of nest complexity in social insects. *Complexity*, 3:15–25.
- Turner, J. (2000a). Architecture and morphogenesis in the mound of *Macrotermes Michaelsoni* in northern namibia. *Cimbebasia*, 16:143–175.
- Turner, J. (2000b). *The Extended Organism. The Physiology of Animal-Built Structures*. Harvard University Press, Cambridge, Massachusetts.

# Ecohydrologic and Biogeochemical Process Networks in Forest Ecosystems in Monsoon East Asia: Identification and Interpretation

Joon Kim<sup>1,2</sup>, Juyeol Yun<sup>1,3</sup>, Jewoo Hong<sup>1</sup>, Hyojung Kwon<sup>1,2</sup>, and Junghwa Chun<sup>4</sup>

<sup>1</sup>Complex Systems Science Laboratory, Department of Landscape Architecture and Rural System Engineering, Seoul National University, Seoul, 151-921 Korea

<sup>2</sup>National Center for AgroMeteorology, Seoul National University, Seoul, 151-921 Korea

<sup>3</sup>Department of Atmospheric Sciences, Yonsei University, 120-749 Korea

<sup>4</sup>Korea Forest Research Institute, Seoul, 130-712 Korea

joon@snu.ac.kr

## Abstract

Forest ecosystems play a critical role in the cycles of carbon and water from local to global scales. These cycles and their variability, in turn, play an important role in the non-trivial emergent and self-organizing interactions between forest ecosystems and their environment. Observational evidence, based on micrometeorological eddy covariance measurements, suggests that heterogeneity and disturbance (both human and natural) in forest ecosystems in monsoon East Asia may facilitate to build resilience for adaptation to change. Yet, the principles that characterize the role of variability in these interactions remain elusive. A process network is defined as a network of feedback loops and the related time scales, which describe the magnitude and direction of the flow of energy, matter, and information between the different variables in a complex system. We attempt to delineate and interpret such process networks by analyzing multivariate ecohydrologic and biogeochemical time series data based on information flow statistics.

## Introduction

Complex systems are systems in which large networks of components with no central control and simple rules of operation give rise to complex collective behavior, sophisticated information processing, and adaptation via learning or evolution (Mitchell, 2009). Thus, the science underlying complex systems should focus not only on the concepts of energy, force and matter, but also on those of feedbacks, information, communication, and purpose.

In Asia, it is of great concern that ecosystem services are being degraded by natural disturbance such as monsoon activity accompanied by typhoons reinforced by anthropogenic factors in a changing climate. Recent finding suggests that under projected climate scenarios, terrestrial carbon sinks in monsoon Asia will decline if the monsoon disturbance will exceed its natural range of variation and if there is no enhancement in the resilience of the ecosystems in this region (e.g., Kwon et al., 2010; Hong and Kim, 2011).

Resilience-based system approach suggests that complex systems evolve through active adaptive cycles to cope with change. Ecohydrologic and biogeochemical processes associated with water and carbon cycles in complex forest

ecosystems can be viewed as a network of processes of a wide range of scales involving various feedback loops.

Finding such networks of feedback loops for key ecosystems in monsoon Asia is of great value and concern. However, the traditional correlation-based analysis cannot delineate such complex processes with detailed information on direction and strength of the coupling between the variables. Following Ruddell and Kumar (2009), we examined the dependence between a series of variables measured at the flux towers in AsiaFlux by quantifying the information flow between the different variables along with the associated time lag. The objective of our study is to test the applicability of information theory to ecohydrologic and biogeochemical systems with the datasets obtained at various forest sites in East Asia with different levels of complexity and heterogeneity.

## Methods and Materials

We used Shannon's information entropy as our methodology (Shannon, 1948) and calculated the transfer entropy ( $TE$ ) to measure the reduction in the entropy of the current state of a measured variable  $X_t^{(j)}$  due to the knowledge of prior state in another variable  $X_t^{(i)}$ , which is in addition to the information provided by the immediate prior history of  $X_t^{(j)}$  (e.g., Ruddell and Kumar, 2009). We normalized  $TE$  using  $m$  (set at 11) discrete bins to estimate the probability distribution function. The information flow process network consists of the asymmetric pair wise  $TE$  between the  $i^{\text{th}}$  and  $j^{\text{th}}$  variable from the set of  $n_v$  observed variables and is represented as an adjacency matrix (Kumar and Ruddell, 2010).

We used the time series data in 2008 from two adjacent KoFlux tower flux sites (in deciduous and coniferous forests) located in Korea. The description of the sites and the data can be found in AsiaFlux homepage (<http://www.asiaflux.net>). In this analysis, we selected 15 variables associated with ecohydrologic and biogeochemical processes in forests, which are atmospheric pressure ( $PA$ ), net ecosystem  $\text{CO}_2$  exchange ( $NEE$ ), gross primary productivity ( $GPP$ ), ecosystem respiration ( $RE$ ), latent heat flux ( $LE$ ), precipitation ( $Precip$ ), solar radiation ( $R_g$ ), air temperature ( $T$ ), vapor pressure deficit ( $VPD$ ), soil temperature ( $T_s$ ), soil water content ( $SWC$ ),

sensible heat flux ( $H$ ), canopy temperature ( $T_c$ ), wind direction ( $WD$ ), and wind speed ( $WS$ ). We computed process networks for each of thirty-six sub-daily time lags between 30 minutes and 18 hours. Our spectral analysis shows that this subdaily time scale explained more than 30% of the variances of the above variables associated with carbon and water cycles, reflecting that this range is an important scale of land-atmosphere interactions. In this process, the complexity and heterogeneity embedded in the observed flux data may hinder the application and interpretation of such information flow statistics. Therefore, estimation and methodological issues were examined by comparing these two adjacent forests with different levels of heterogeneity and complexity.

## Preliminary Results

The adjacency matrix for the 15 variables results in 210 potential pairwise couplings, about 25% out of which were found to be statistically significant at one or more time lags for both deciduous and coniferous forests. Preliminary results on network matrix are presented in Tables 1-4.

Table 1. Network matrix for mutual information

Atz	PA	NEE	GPP	RE	LE	Precip	Rg	T	VPD	Ts	SWC	H	Tc	WD	WS
PA	<i>85.5 (94)</i>	2.7 (3.4)	2.9 (3.5)	12.3 (8.9)	1.9 (2.8)	1.8 (1.8)	4.7 (4.8)	11.8 (8.5)	6.7 (6.0)	16.3 (-)	14.2 (13.7)	3.8 (5.4)	10.7 (9.9)	5.1 (3.8)	4.9 (6.2)
NEE	x (0)	0.1 (0.1)	0.1 (0.1)	x (0)	0.7 (1)	1.2 (1.3)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (2.8)	x (0)
GPP	x (0)	0.2 (0.2)	0.1 (0.1)	x (0)	0.7 (1)	1.1 (1.1)	x (0)	x (0)	x (0)	x (-)	x (0)	0.7 (0.7)	x (0)	x (2.2)	x (0)
RE	x (0)	x (1.3)	x (2.7)	x (0.1)	2.2 (4)	0.8 (0)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (0)	x (0)
LE	x (0)	0.7 (1)	0.6 (0.8)	x (0)	0.1 (0.1)	1.9 (1.3)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (0)	x (0)
Precip	x (0)	1.5 (0)	1.7 (0)	x (0)	2.3 (0)	0.1 (0.1)	x (0)	x (0)	x (0)	x (-)	1.2 (0)	3.2 (0)	x (0)	x (0)	x (0)
Rg	x (0)	0.7 (0.9)	0.7 (0.8)	x (1.2)	0.7 (0.8)	1.2 (1)	x (0)	x (0)	x (0)	x (-)	x (0)	0.3 (0)	x (0)	x (1.7)	x (0)
T	x (0)	3.4 (0)	x (0)	x (0.2)	x (0)	0.8 (1.1)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (0)	x (0)
VPD	x (0)	2.1 (0)	1.7 (1.3)	x (0)	1.3 (1.1)	1.2 (1)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (3.2)	x (0)
Ts	x (-)	3.5 (-)	x (-)	x (-)	2.5 (-)	1.3 (-)	x (-)	x (-)	x (-)	x (-)	x (-)	x (-)	x (-)	x (-)	x (-)
SWC	x (0)	x (2.2)	x (2.2)	x (0)	x (1.6)	0.8 (0.6)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (2.9)	x (0)
H	x (0)	0.8 (0.8)	0.8 (0.8)	x (1.3)	0.9 (0.8)	2.3 (1.8)	x (0)	x (0)	x (0)	x (-)	x (0)	0.1 (0)	x (0)	x (0)	x (0)
Tc	x (0)	2.1 (1.9)	x (0)	x (0.2)	1.4 (0)	1.0 (0)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (0)	x (0)
WD	x (0)	2.7 (2.1)	2.6 (1.9)	x (0)	2.2 (1.2)	1.8 (1.3)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (0.1)	x (0)
WS	x (0)	x (0)	x (0)	x (2.1)	x (0)	1.8 (0)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (0)	x (0)

information flow on the interval, including the first significant lag, last significant lag, number of significant lags, and peak time lag. Significant lag times are [first-last (number), max].

Table 3. Network matrix for the ratio of the maximum lag to mutual information

Atz	PA	NEE	GPP	RE	LE	Precip	Rg	T	VPD	Ts	SWC	H	Tc	WD	WS
PA	x (0)	x (0)	x (0)	x (0)	2.7 (0)	1.9 (0)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (2.8)	x (0)
NEE	x (0)	0.1 (0.1)	0.1 (0.1)	x (0)	0.7 (1)	1.2 (1.3)	x (0)	x (0)	x (0)	x (-)	x (0)	0.7 (0.7)	x (0)	x (2.2)	x (0)
GPP	x (0)	0.2 (0.2)	0.1 (0.1)	x (0)	0.7 (1)	1.1 (1.1)	x (0)	x (0)	x (0)	x (-)	x (0)	0.7 (0.7)	x (0)	x (2.2)	x (0)
RE	x (0)	x (1.3)	x (2.7)	x (0.1)	2.2 (4)	0.8 (0)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (0)	x (0)
LE	x (0)	0.7 (1)	0.6 (0.8)	x (0)	0.1 (0.1)	1.9 (1.3)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (0)	x (0)
Precip	x (0)	1.5 (0)	1.7 (0)	x (0)	2.3 (0)	0.1 (0.1)	x (0)	x (0)	x (0)	x (-)	1.2 (0)	3.2 (0)	x (0)	x (0)	x (0)
Rg	x (0)	0.7 (0.9)	0.7 (0.8)	x (1.2)	0.7 (0.8)	1.2 (1)	x (0)	x (0)	x (0)	x (-)	x (0)	0.3 (0)	x (0)	x (1.7)	x (0)
T	x (0)	3.4 (0)	x (0)	x (0.2)	x (0)	0.8 (1.1)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (0)	x (0)
VPD	x (0)	2.1 (0)	1.7 (1.3)	x (0)	1.3 (1.1)	1.2 (1)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (3.2)	x (0)
Ts	x (-)	3.5 (-)	x (-)	x (-)	2.5 (-)	1.3 (-)	x (-)	x (-)	x (-)	x (-)	x (-)	x (-)	x (-)	x (-)	x (-)
SWC	x (0)	x (2.2)	x (2.2)	x (0)	x (1.6)	0.8 (0.6)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (2.9)	x (0)
H	x (0)	0.8 (0.8)	0.8 (0.8)	x (1.3)	0.9 (0.8)	2.3 (1.8)	x (0)	x (0)	x (0)	x (-)	x (0)	0.1 (0)	x (0)	x (0)	x (0)
Tc	x (0)	2.1 (1.9)	x (0)	x (0.2)	1.4 (0)	1.0 (0)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (0)	x (0)
WD	x (0)	2.7 (2.1)	2.6 (1.9)	x (0)	2.2 (1.2)	1.8 (1.3)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (0.1)	x (0)
WS	x (0)	x (0)	x (0)	x (2.1)	x (0)	1.8 (0)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (0)	x (0)

Table 4. Network matrix for time lags of significant information flow on the interval

Tau	PA	NEE	GPP	RE	LE	Precip	Rg	T	VPD	Ts	SWC	H	Tc	WD	WS
PA	x (0)	x (0)	x (0)	x (0)	1-1310332 (0)	1-7407 (0)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (-7-717)	x (0)
NEE	x (0)	2-873 (-9-902)	2-903 (-9-902)	x (0)	1-10992 (-8-908)	1-10992 (-8-908)	x (0)	x (0)	x (0)	x (-)	x (0)	1-7351 (-2-124)	x (0)	1-7351 (-2-124)	x (0)
GPP	x (0)	2-873 (-9-902)	2-903 (-9-902)	x (0)	1-10992 (-8-908)	1-10992 (-8-908)	x (0)	x (0)	x (0)	x (-)	x (0)	1-7351 (-2-124)	x (0)	1-7351 (-2-124)	x (0)
RE	x (0)	x (-1-33525)	x (-25-36632)	x (-2-6520)	36-362136 (-36-36430)	1-5461 (0)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (0)	x (0)
LE	x (0)	1-4002 (-8-908)	1-4002 (-8-908)	x (0)	2-972 (-2-11012)	1-360736 (-2-11012)	x (0)	x (0)	x (0)	x (-)	x (0)	x (0)	x (0)	x (0)	x (0)
Precip	x (0)	1-362101 (-9-908)	1-360931 (-9-908)	x (0)	1-360931 (-9-908)	1-360931 (-9-908)	x (0)	x (0)	x (0)	x (-)	x (0)	1-12111 (-33-36636)	x (0)	x (-1-4461)	x (0)
Rg	x (0)	1-362101 (-9-908)	1-360931 (-9-908)	x (0)	1-360931 (-9-908)	1-360931 (-9-908)	x (0)	x (0)	x (0)	x (-)	x (0)	1-12111 (-33-36636)	x (0)	x (-1-4461)	x (0)
T	x (0)	1-362101 (-9-908)	1-360931 (-9-908)	x (0)	1-360931 (-9-908)	1-360931 (-9-908)	x (0)	x (0)	x (0)	x (-)	x (0)	1-12111 (-33-36636)	x (0)	x (-1-4461)	x (0)
VPD	x (0)	1-362101 (-9-908)	1-360931 (-9-908)	x (0)	1-360931 (-9-908)	1-360931 (-9-908)	x (0)	x (0)	x (0)	x (-)	x (0)	1-12111 (-33-36636)	x (0)	x (-1-4461)	x (0)
Ts	x (0)	1-362101 (-9-908)	1-360931 (-9-908)	x (0)	1-360931 (-9-908)	1-360931 (-9-908)	x (0)	x (0)	x (0)	x (-)	x (0)	1-12111 (-33-36636)	x (0)	x (-1-4461)	x (0)
SWC	x (0)	1-362101 (-9-908)	1-360931 (-9-908)	x (0)	1-360931 (-9-908)	1-360931 (-9-908)	x (0)	x (0)	x (0)	x (-)	x (0)	1-12111 (-33-36636)	x (0)	x (-1-4461)	x (0)
H	x (0)	1-362101 (-9-908)	1-360931 (-9-908)	x (0)	1-360931 (-9-908)	1-360931 (-9-908)	x (0)	x (0)	x (0)	x (-)	x (0)	1-12111 (-33-36636)	x (0)	x (-1-4461)	x (0)
Tc	x (0)	1-362101 (-9-908)	1-360931 (-9-908)	x (0)	1-360931 (-9-908)	1-360931 (-9-908)	x (0)	x (0)	x (0)	x (-)	x (0)	1-12111 (-33-36636)	x (0)	x (-1-4461)	x (0)
WD	x (0)	1-362101 (-9-908)	1-360931 (-9-908)	x (0)	1-360931 (-9-908)	1-360931 (-9-908)	x (0)	x (0)	x (0)	x (-)	x (0)	1-12111 (-33-36636)	x (0)	x (-1-4461)	x (0)
WS	x (0)	1-362101 (-9-908)	1-360931 (-9-908)	x (0)	1-360931 (-9-908)	1-360931 (-9-908)	x (0)	x (0)	x (0)	x (-)	x (0)	1-12111 (-33-36636)	x (0)	x (-1-4461)	x (0)

Table 2. Network matrix for uncertainty percentage

Atz	PA	NEE	GPP	RE	LE	Precip	Rg	T	VPD	Ts	SWC	H	Tc	WD	WS
PA	<i>100 (100)</i>	4.9 (5.7)	5.2 (6.1)	13.2 (14.2)	3.9 (4.4)	12.1 (11.7)	7.9 (8.1)	12.5 (10.4)	8.2 (7.8)	17 (-)	21.6 (23.9)	5.9 (8.3)	11.4 (10.8)	5.8 (4.8)	6.3 (8.3)
NEE	2.9 (3.6)	<i>100 (100)</i>	11.5 (12.7)	2.6 (3)	14.7 (10.9)	7.9 (8)	18.6 (16.8)	2 (3)	3.8 (5.6)	21 (-)	3.3 (3.9)	12.3 (13.1)	3.2 (3.9)	2.7 (4.3)	3.1 (4.8)
GPP	3.1 (3.7)	79.6 (79.5)	<i>100 (100)</i>	21 (3)	15.5 (11.5)	9 (9.5)	19.4 (18.5)	2.2 (2.9)	4.8 (5.0)	23 (-)	3.1 (4.4)	12.4 (13.9)	3.4 (3.6)	2.8 (5)	3.1 (3.3)
RE	13.2 (7.3)	3.3 (5.1)	3.5 (2.8)	<i>100 (100)</i>	5.1 (2.8)	15.2 (5.1)	6.9 (5)	97.7 (26.2)	19.4 (3)	19.4 (-)	12.5 (3.5)	4.8 (3.8)	48.5 (25.2)	3.3 (7)	4.4 (2.3)
LE	2.1 (3)	13.1 (4.6)	13.6 (12.7)	2.7 (3)	<i>100 (100)</i>	4.4 (7.7)	13.3 (16.6)	3.2 (3.9)	5.3 (8.2)	24 (-)	3.8 (6.1)	8.4 (4.1)	3.8 (4.4)	2.6 (2.8)	1.9 (3.8)
Precip	2 (2)	2.2 (2.1)	2.4 (2.6)	2.5 (3.7)	1.4 (1.9)	<i>100 (100)</i>	2.6 (2.5)	2.5 (3.3)	2 (2.1)	1.6 (-)	3.8 (5.8)	1.5 (1.6)	2 (2.6)	1.3 (1.5)	1.1 (2.4)
Rg	5.1 (5.1)	20.2 (16.7)	20.7 (19.1)	4.5 (6.1)	16.2 (15.5)	10.3 (9.2)	<i>100 (100)</i>	4.2 (5.6)	7.9 (10.3)	4 (-)	4.9 (8.3)	31 (33.2)	6.7 (8.8)	4.7 (5.8)	3.3 (8.4)
T	12.6 (10.1)	3.4 (4.9)	3.7 (4.5)	58.5 (49.3)	6.1 (5.6)	15.3 (1.3)	6.6 (8.6)	<i>100 (100)</i>	22.6 (20.4)	23.1 (-)	12.9 (12.2)	4.8 (5.4)	58.9 (59.8)	3.3 (3.7)	4 (4.7)
VPD	7.2 (6.4)	5.6 (7.2)	6.6 (7.6)	17 (20.4)	8.7 (9.9)	10.7 (10.4)	10.8 (13.3)	10.6 (17.2)	<i>100 (100)</i>	10.1 (-)	10.7 (7.9)	6.7 (9.5)	19.9 (17.7)	3.2 (3.8)	4.4 (9.7)
Ts	17.5 (-)	3.7 (-)	3.9 (-)	20 (-)	4.6 (-)	10.4 (-)	6.4 (-)	23.5 (-)	11.8 (-)	<i>100 (-)</i>	21.1 (-)	5.4 (-)	20.9 (-)	4.5 (-)	6.8 (-)
SWC	15.2 (16.7)	3.9 (4.4)	3.7 (5)	8.8 (7.1)	5.1 (6.3)	16.3 (23.2)	5.3 (9.2)	9 (8.8)	8.6 (6.8)	14.5 (-)	<i>100 (100)</i>	4.3 (7.8)	8 (8.5)	4.4 (3.8)	4.3 (6.8)
H	4 (5.8)	14.2 (14.4)	14.1 (15.8)	3.3 (4.8)	10.8 (14.5)	6.2 (6.4)	32.9 (16.6)	3.2 (4.6)	5.3 (8.1)	3.6 (-)	4.1 (7.8)	<i>100 (100)</i>	5.3 (5.2)	5.1 (7.1)	3.1 (7.8)
Tc	11.4 (10.6)	5.4 (6.1)	5.6 (5.9)	48.6 (48.1)	7.1 (8.5)	12.5 (15.1)	10.3 (10.6)	58.1 (60.8)	22.7 (21.3)	20.2 (-)	11.3 (7.2)	7.7 (7.4)	<i>100 (100)</i>	3.2 (4)	4.5 (5.1)
WD	5.4 (6.1)	4.3 (5.8)	4.4 (6.3)	31 (3.2)	4.6 (3.5)	7.8 (7.5)	6.7 (8)	3 (3.2)	3.4 (3.8)	4.1 (-)	5.9 (4.6)	3 (3.4)	3 (3.4)	<i>100 (100)</i>	4.1 (5.3)
WS	5.3 (6.4)	4.4 (6.1)	4.3 (7)	3.7 (3.8)	3 (6.9)	5.8 (11.5)	4.3 (10.6)	3.3 (3.9)	4.2 (9.5)	5.5 (-)	5.1 (7.7)	3.7 (9)	3.8 (4.1)	3.6 (5.3)	<i>100 (100)</i>

Table 1 shows the matrix for the mutual information between pairs of variables at zero time lag. Source variable  $X$  index  $i$  is in rows; sink variable  $Y$  index  $j$  is in columns. Matrix is symmetric. Italics indicate matrix diagonal. All values are in percent. The values before and with parenthesis are for deciduous (GDK) and coniferous forest (GCK), respectively.

Table 2 shows the matrix for the percentage of uncertainty of each  $Y$  explained by  $X$ . Table 3 shows the matrix for the ratio of the maximum lag to mutual information for all significant couplings. Table 4 shows time lags of significant

## Acknowledgments

This study was supported by Long-Term Ecological Study and Monitoring of Forest Ecosystem Project of Korea Forest Research Institute, a grant (code 1-8-3) from Sustainable Water Resource Research Center for 21<sup>st</sup> Century Frontier Research Program; A3 Foresight Program of Korea Research Foundation; and Research Settlement Fund for the new faculty of Seoul National University. We thank Minseok Kang and Bindu Malla Thakuri and Jaehi Yoo for their field support and data analysis.

## Selected References

- Hong, J. and Kim, J. (2011). Impact of the Asian monsoon climate on ecosystem carbon and water exchanges: a wavelet analysis and its ecosystem modeling implications. *Global Change Biology*, doi: 10.1111/j.1365-2486.2010.02337.x
- Kumar, P. and Ruddell, B. L. (2010). Information driven ecohydrologic self-organization. *Entropy*, 12, 2085-2096.
- Kwon, H., Kim, J., Hong, J., and Lim, J.-H. (2010). Influence of the Asian monsoon on net ecosystem carbon exchange in two major ecosystems in Korea. *Biogeosciences*, 7, 1493-1504.
- Mitchell, M. (2009). *Complexity: A guided tour*. Oxford University Press.
- Ruddell, B. L. and Kumar, P. (2009). Ecohydrologic process networks: 1. Identification. *Water Resource Research*, 45, W03419, doi:10.1029/2008WR007279, 2009
- Shannon, C. E. (1948). A mathematical theory of communication, *Bell Syst. Tech. J.* 27, 379-423.

# Parsimonious Modeling of Scaling Laws in Genomes and Transcriptomes

Carole Knibbe<sup>1,3</sup>, David P. Parsons<sup>2,3</sup> and Guillaume Beslon<sup>2,3</sup>

<sup>1</sup>Université de Lyon, CNRS, INRIA, Université Lyon 1, LIRIS, UMR5205, F-69622, France

<sup>2</sup>Université de Lyon, CNRS, INRIA, INSA-Lyon, LIRIS, UMR5205, F-69621, France

<sup>3</sup>IXXI, Institut Rhône-Alpin des Systèmes Complexes, Lyon, F-69007, France  
carole.knibbe@liris.cnrs.fr

## Abstract

We report here the use of Aevol, a software developed in our team to unravel the indirect selective pressures (i.e. pressures for robustness and/or evolvability) that act on the genome and transcriptome structures. Using Aevol, we have shown that these structures are under strong – although indirect – pressure due to the mutagenic effect of chromosomal rearrangements. Individuals undergoing high spontaneous rearrangement rates show more compact structures than individuals undergoing lower rates. This phenomenon concerns genome size and content (non-coding DNA, presence of operons, number of genes) as well as gene network (number of nodes and links) thus reproducing parsimoniously a large panel of known biological properties. The results reported here have been published in *Mol. Biol. Evol.* (Knibbe et al., 2007), *Biosystems* (Beslon et al., 2010) and *Alife XII* (Parsons et al., 2010).

## Introduction

Largescale comparative analysis of sequenced genomes has revealed that several molecular traits follow characteristic scaling laws. For instance, the genome size has been shown to scale as a power-law of the spontaneous mutation rate in DNA-based microbes (Drake, 1991). More recently, different genomic properties have been shown to follow power-law distributions (Luscombe et al., 2002). In prokaryotes for instance, it was shown that the number of genes in each functional category scales as a power-law of the total number of genes and that the exponent of this law depends on the functional role of the family: The number of transcription factors, in particular, scales quadratically with the total number of genes while metabolic genes scale linearly (van Nimwegen, 2003). This increase is also correlated with the size of the genome (Konstantinidis and Tiedje, 2004).

The origins of such scaling laws remain an open question. Actually, despite the tremendous advance in the fields of genomics and transcriptomics, it is still not clear whether these “molecular allometric laws” result from selective constraints (e.g., selection for short genomes or integrated networks) or from the neutral dynamics of the evolutionary process.

An original approach to study the origins of genomic structures is to use *in silico* models of evolution. In such

models, the evolutionary forces are precisely tuned and it is possible to test experimentally how they shape the organisms’ structure. *In silico* evolution has already shown that darwinian evolution can have counter-intuitive effects, due to indirect selective pressures. For example, using the avida framework, Wilke et al. (2001) have shown that the long-term survival of a lineage not only depends on its fitness, but also on its mutational robustness. However, most digital genetic frameworks lack a precise description at the molecular level. That is why we have developed Aevol (“Artificial Evolution”) and its extension R-Aevol (“Regulation in Aevol”). It specifically focuses on the molecular level in order to unravel the evolutionary pressures that act on genomes and transcriptomes. We report here the main results we got with Aevol. These results have been published in *Molecular Biology and Evolution* (Knibbe et al., 2007), *Biosystems* (Beslon et al., 2010) and *Alife XII* (Parsons et al., 2010). Aevol is freely available upon request from the authors.

## The Aevol model

In Aevol, organisms own a circular, double-stranded genome of binary “nucleotides”. Predefined signaling sequences as well as an artificial genetic code allow to detect the coding sequences and to translate them into abstract “proteins”. We defined an artificial chemistry that describes the metabolism in a mathematical language: We assume that there is a one-dimensional space of all possible metabolic functions in which proteins are represented by a subset describing their metabolic contribution. This subset is described by parameters encoded in the coding sequence of the protein. Mutations in this sequence change these parameters, hence the metabolic activity of the protein.

In Aevol, the transcription rate of a given gene depends only on its own promoter sequence. In R-Aevol, proteins may have a regulatory activity besides their metabolic activity, thus being able to enhance or inhibit the transcription of other genes by binding to their promoters. The resulting transcription level is used to scale up or down both the metabolic and the regulatory activities of the protein. Due to this regulatory process, the transcription levels of the genes



vary during the organism life and so do the protein activities.

In Aevol as well as in R-Aevol, the global metabolism is computed by combining all the proteins' activities and the phenotype represents the degree of realization of each possible metabolic function. The fitness of the organism is then computed as the distance between the phenotype and a pre-defined target. The fittest organisms are allowed to replicate, with small mutations and large rearrangements (duplications, deletions, inversions, translocations) occurring randomly during the replication. Thus the genome size, gene number and gene order are free to evolve. In R-Aevol, mutations and rearrangements can also modify the regulatory network by either duplicating/deleting genes or promoter regions or by modifying their binding potentials.

## Results

Digital genetics models are experimental models: Population of individuals evolve in different conditions and, by observing the genomic and transcriptomic structures of the evolved organisms, one then links the structures to the evolutionary conditions. Analysis of the lineages then enables to unravel the origins of the observed structures, ideally by discovering invariant properties in all simulations. In Aevol, we classically explore the influence of mutation rates, rearrangement rates and selection strength. The most striking results were obtained by exploring the influence of rearrangement rates on the different organization levels of the model:

I) By observing the genome length of evolved organisms, we observed a linear scaling between the rearrangement rate and the length of the non-coding sequences in Aevol's genomes. We have shown that this scaling is due to an indirect selective pressure acting on the non-coding sequences: Due to chromosomal rearrangements, non-coding sequences have a mutagenic effect on the surrounding genes. This long-term selective pressure offers a new explanation to variability of genome size and content (Knibbe et al., 2007).

II) By reproducing in R-Aevol the same experiment, we have shown that this pressure also acts at the transcriptomic level. Regulation networks evolved under different rearrangement rates show huge structural differences, ranging from very small hardly connected networks (high rates) to large and densely connected ones (low rates). Moreover, like in prokaryotes, the number of transcription factors scales quadratically with the number of genes (Beslon et al., 2010).

III) Finally, we showed that this indirect pressure induces many side effects. In particular, under high rearrangement rates, genome compaction causes a fusion of transcribed sequences favouring operons (Parsons et al., 2010).

Thus, by changing a single parameter in the simulations – the spontaneous rearrangement rate – we were able to reproduce genomic and transcriptomic structures ranging from virus-like structures to prokaryote-like and, ultimately, eukaryote-like ones. Moreover, we were able to show that the best final organisms obtained in all these simulations

share the same variability level. If we measure the probability for the best final organisms to reproduce neutrally (i.e. the product of its offspring number  $W$  by its fraction of neutral offspring  $F_\nu$ ), we always observe that  $F_\nu W \sim 1$ , showing that these very different organisms all share a same exploration-exploitation compromise, an evident hallmark of indirect selection.

Of course, in Aevol many selective and non-selective effects have been neglected (energetic costs, mutational biases...) that may interact with the indirect selective pressure we isolated. However, in the model, this indirect selective pressure appears to be strong enough to overcome direct selective pressure (high rearrangement rates forbidding organisms to increase their gene repertoire). Thus it is likely to have an effect in real organisms. We now use Aevol to better understand the traces that indirect selection may leave in genomes. We will then be able to search for these traces in the sequences that accumulate in databases.

## References

- Beslon, G., Parsons, D. P., Sanchez-Dehesa, Y., Peña, J.-M., and Knibbe, C. (2010). Scaling laws in bacterial genomes: A side-effect of selection of mutational robustness. *BioSystems*, 102(1):32–40.
- Drake, J. W. (1991). A constant rate of spontaneous mutation in dna-based microbes. *Proc Natl Acad Sci USA*, 88(16):7160–7164.
- Knibbe, C., Coulon, A., Mazet, O., Fayard, J.-M., and Beslon, G. (2007). A long-term evolutionary pressure on the amount of noncoding DNA. *Mol. Biol. Evol.*, 24(10):2344–2353.
- Konstantinidis, K. T. and Tiedje, J. M. (2004). Trends between gene content and genome size in prokaryotic species with larger genomes. *Proc Natl Acad Sci USA*, 101(9):3160–3165.
- Luscombe, N. M., Qian, J., Zhang, Z., Johnson, T., and Gerstein, M. (2002). The dominance of the population by a selected few: power-law behaviour applies to a wide variety of genomic properties. *Genome Biol*, 3(8):RE-SEARCH0040.
- Parsons, D. P., Knibbe, C., and Beslon, G. (2010). Importance of the rearrangement rates on the organization of transcription. In *Proceedings of Artificial Life XII, MIT Press*, pages 479–486.
- van Nimwegen, E. (2003). Scaling laws in the functional content of genomes. *Trends Genet*, 19(9):479–484.
- Wilke, C. O., Wang, J. L., Ofria, C., Lenski, R. E., and Adami, C. (2001). Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, 412(6844):331–333.

# Scalable Co-Evolution of Soft Robot Properties and Gaits

Davis Knox<sup>1</sup> and John Rieffel<sup>1</sup>

<sup>1</sup>Union College, Schenectady, NY 12308  
rieffelj@cs.union.edu

## Abstract

One of the outstanding challenges in soft robotics is the chicken-and-egg problem of body/brain design: generation of locomotion is predicated on the existence of a locomotion-capable body, and generation of body plans is predicated upon the existence of effective locomotion. This problem is compounded by the high degree of coupling between the *material properties* of a soft body (such as stiffness or damping coefficients) and the effectiveness of a gait. In this work we describe a means by which the material properties of a simulated soft body co-evolve alongside locomotive gaits. Improvements in simulation time, with no loss of overall fitness, are obtained by incrementally increasing mesh density over the course of evolution.

## Introduction

Imagine a soft, resilient and deformable robot able to change shape and squeeze through small apertures. The idea of using such a robot for urban search and rescue holds great appeal, particularly in light of recent tragic earthquakes in China, New Zealand, and Japan. Once the domain of science fiction, soft robots are approaching reality – thanks to recent advances in engineering and material science. Unfortunately, the very properties which make soft robots so appealing also introduce significant obstacles, especially in the domains of design and control. Elasticity and deformability come at the cost of resonances and tight dynamic coupling between components (Trimmer, 2007) – properties which are often assiduously avoided in conventional engineering approaches to robotic design. Small changes to the elasticity of a soft robot can cause unexpectedly large changes in performance.

Absent the analytical design methodologies available to conventional rigid robots, one compelling approach lies in *evolutionary design*, a field which has had considerable success in other complex design domains ranging from satellite antennae (Lohn et al., 2005) to telescope lenses (Al-Sakran et al., 2005) to elaborate tensegrity structures (Rieffel et al., 2009b).

The problems of soft robot design can be summarized with three questions:

- What should a soft robot *look like*? (*Morphology*)
- What physical properties should a soft robot have? (*Material*)
- How should a robot *move*? (*Locomotion*)

Of course, these are not independent variables: solving each problem is predicated upon, and sensitive to, the pre-existence of solutions to the corresponding problems. The design of a soft robot's locomotive gait, for instance, depends upon both its morphology and properties such as elasticity and friction.

This is in a sense an elaboration on the chicken-and-egg problem posed by body/brain design in more conventional robots (Pollack et al., 1999, 2001), with the added complexities which come from the effects of material properties upon a soft body's dynamics. In light of that, our approach to solving the problem will be similar: co-evolution. Earlier work has focused on co-evolving soft robot morphology with gaits (Rieffel et al., 2009a; Rieffel, 2010), and so this research focuses on the related problem of co-evolving material properties alongside gaits.

This paper describes how, given a specific soft robot's *shape*, we are able arrive at effective locomotion by co-evolving gaits – muscle firing patterns – alongside finely tuned material properties such as stiffness and damping coefficient. In doing so, we demonstrate a connection between material properties and gaits. Furthermore, in order to address the computational overhead imposed by soft body simulation, we introduce a method which scales model mesh resolution over the course of evolution, such that a large early portion of evolutionary time is devoted to low resolution models of the robot, and as evolution progresses mesh resolution increases. This resolution scaling achieves fitnesses comparable to those achieved by fixed high resolution while reducing overall computation time.

## Simulating Soft Robots

Once the domain of Finite Element Analysis (FEA) and Computational Fluid Dynamics (CFD), physics simulation is now much more accessible thanks to recent advances in



commercial off-the-shelf video-game physics engines accelerated by massively parallel graphics cards (GPUs). This General Purpose Computing on Graphics Processing Units (GPGPU) can provide speedups of several orders of magnitude over software-only simulation. (Banzhaf and Harding, 2009). In particular, our research uses NVidia's PhysX engine because of its ability to simulate complex three-dimensional *soft bodies*.

### Soft Bodies in PhysX

Soft bodies in PhysX are represented as tetrahedral meshes, where single tetrahedra (Figure 1) are connected to their neighbors at their common vertices. The material properties of a soft body mesh can be tuned by varying a set of constraints placed upon the tetrahedra within a mesh. Two values, stretching stiffness and damping co-efficient, tune the parameters of a spring-and-damper system along each edge of the tetrahedron. A tetrahedral mesh with high stretching stiffness will try hardest to maintain its shape, while one with a low stiffness will flop to the floor like a deflating balloon. The damping coefficient of a soft body changes how fast it returns to equilibrium after a perturbation. A low damping co-efficient allows soft bodies in motion to oscillate more. A third constraint, volume stiffness, determines how hard each tetrahedra attempts to maintain a constant volume – a mesh with low volume stiffness will resemble a flat puddle more than a balloon. Changing each of these values affects the softness of all tetrahedra in a soft body, although not necessarily in a linear manner. As illustrated by Figure 3, by varying these material properties, the behavior of soft bodies in PhysX can range from a near fluid, to rubbery Jell-O to a semi-rigid plastic. Finally, we also chose to vary the friction of the crawling surface, within a relatively narrow range, in order to be able to change how well the soft material gripped the substrate.

The bottleneck for soft bodies simulation is the density of the underlying tetrahedral mesh: simulation slows dramatically as the number of tetrahedra in a mesh grow (Figure 2). The trade-off is that low-resolution meshes, by modeling fewer nuances of the soft body, such as body wall folding, risk having lower fidelity to the real-world behavior of the corresponding soft body.

**Soft Body Gaits** One of the more interesting consequences of soft robotics is the lack of conventional actuators. Because suppleness and deformability are important, devices like servos and stepper motors are not viable. Absent those, one valuable alternative is nitinol “memory wire” (Trimmer, 2007). These artificial muscles act essentially as linear actuators, and can be modeled as applying equal and opposite force vectors to their two attachment points.

Given a fixed set of muscles in a soft robot, a simple way to represent their firing patterns is through a square wave characterized by a duty cycle, a phase offset, and a period

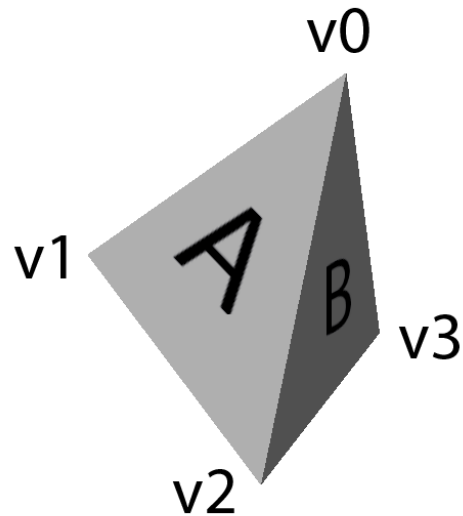


Figure 1: Soft bodies in PhysX are built out of tetrahedral meshes. Each tetrahedron is defined by four vertices and four corresponding faces. The material properties of a mesh can be tuned by changing the stretching and damping coefficients of spring-and-dampers systems along the edge, and by changing the tetrahedron's resistance to volume changes.

(Figure 4). The period of the firing pattern represents the time between the square wave's rising edges. Duty cycle corresponds to the percent of time that a muscle is “on” during that period. Finally, the phase of the firing pattern represents the delay before the first rising edge.



Figure 4: Soft robot gaits are composed of firing patterns for a set eight symmetrical muscles (four per side). Each of the eight patterns is described by a unique duty cycle, phase, and period.

Figure 5 shows the layout of the eight muscles on our model robot. It is worth emphasizing that although the muscle placement is bilaterally symmetric, in our genetic algorithm we place no such constraints on the eight matching firing patterns.

### Co-Evolving Gaits and Material

Our goal was to simultaneously discover a suitably matched gait/property pair, and so it seemed natural to gauge fitness by the linear distance traveled by the soft body over a fixed

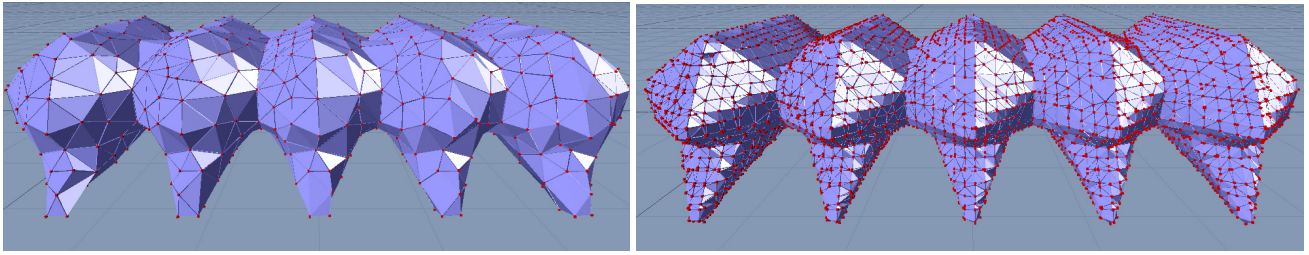


Figure 2: The same robot model with low mesh resolution (left) and high resolution (right).

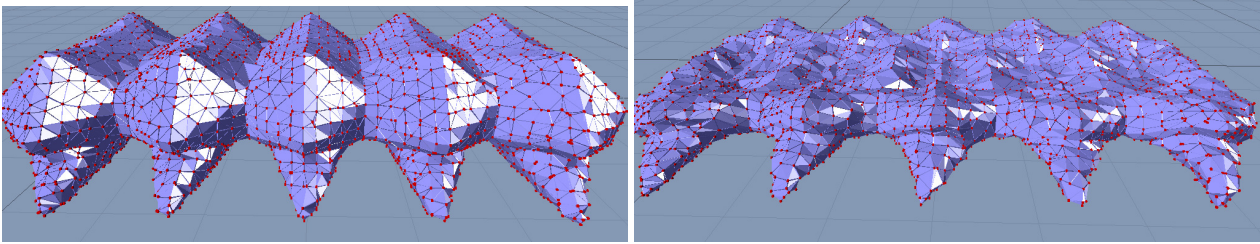


Figure 3: Changing the underlying material properties can drastically affect both the shape and the behavior of a soft body. Images of the same soft body with high (left) and low (right) stretching stiffnesses.

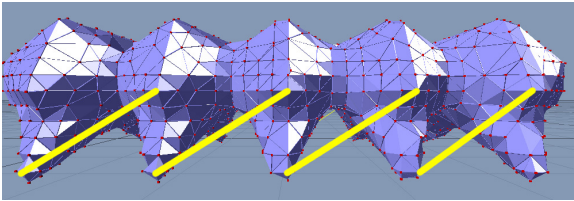


Figure 5: An illustration of the linear actuator “muscles” of the simulated soft body. Although muscles are aligned with bilateral symmetry, no symmetry constraints were placed on the underlying firing patterns.

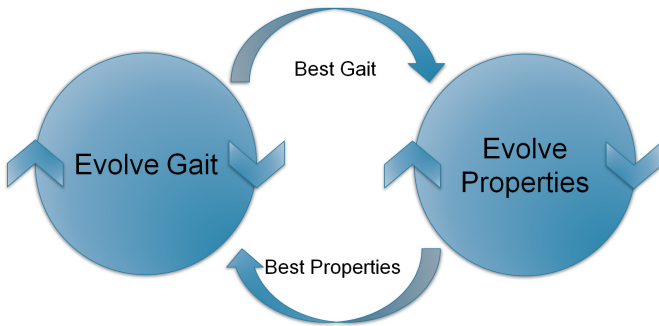


Figure 6: Evolution occurs in two parallel populations: the population of gaits uses the current-best set of material properties, while the set of population of properties uses the current best gait.

number (8000) of simulator time steps. However, the fitness of a specific gait can vary greatly depending upon the underlying material properties, and, similarly, the fitness of a material property set depends greatly upon the gait it is tested against.

Our solution was to co-evolve a population of gaits in lock-step with a population of material properties, each with a common fitness function (but separate fitness values). Figure 6 illustrates our approach. One population was composed of soft body gaits, where a gait genome is composed of eight phase/duty/period tuples, one for each muscle. No other information, such as muscle location, is encoded in the genome.

Values for the properties were limited to keep results realistic. Ranges are as follows (note that in PhysX, like most physics simulators, these properties are unit-less):

Property	Min	Max
Volume Stiffness	0.1	1.0
Stretching Stiffness	0.3	1.0
Friction	0.5	1.0
Damping	0.0	1.0

Initially, a fixed “best guess” of material property values (those which were used to produce the results of our earlier work in Rieffel (2010)) was used for evaluating the fitness of each gait. The second population evolved soft body material properties, where a single genome contained values for a specific set of stretching stiffness, volume stiffness, damping co-efficient and body friction. Initially, for this population’s fitness evaluations, a fixed “best guess” of firing patterns –

in this case arrived at via human trial and error – was used.

Every tenth generation, the gait used for material property fitness evaluation was updated with the current highest-fitness gait from the gait population, and the material properties used for gait population evaluation were updated from the highest-fitness property values.

With population sizes of 40, a typical run took approximately 24 hours to evaluate 100 generations of each population on a 2.66 GHz Core i7 processor with 6 GB of RAM.

## Results and Discussion

We have two analyses of our experiments to offer. The first is a more qualitative description of the gaits produced by our system and some insight into how changing material properties affect fitness. The second more quantitatively explores the effect of scaling the mesh resolution over the course of an evolutionary run.

### Gaits and Material Properties

Our experiments consistently produced interesting and effective gaits, and this suggests that being able to change material properties alongside firing patterns has a positive effect upon the outcome. A qualitative and visual representation lies in videos of the actual gaits, which can be seen at the following URLs:

<http://www.youtube.com/watch?v=dOV33dHRaD8>

<http://www.youtube.com/watch?v=1fJ4IHni5p0>

The first video shows a bilaterally asymmetric gait. Firing patterns on each side of the body co-ordinate in a rough front-to-back wave pattern in order to collectively lift the limbs upwards and forwards during the upswing, before relaxing into the downswing to pull against the ground. The relative softness of the material can be seen in the amount of flexing undergone by each leg. The second video, by comparison shows a more symmetrical gait achieved by a forward-moving wave which produces what almost looks like a gallop.

There were some distinct differences in material property values across these two runs, as summarized below:

Property	Bipedal	Wave
Volume Stiffness	0.986	0.996
Stretching Stiffness	0.982	0.998
Friction	0.598	0.804
Damping	0.0004	0.0

The most notable difference is the friction – corresponding to the stickiness of the robot’s feet, however when watching the videos, the relatively minor numerical differences in the other property values appear, at least qualitatively, to be reflected in the behavior of the soft bodies.

Of further interest is the change in best-of material values properties which occur over the course of an evolutionary run, as shown in Figure 8. While damping coefficient and

volume stiffness show relatively monotonic progress toward a fixed value, stretching stiffness and friction vary consistently across a relatively wide range during evolution. The effect of material property changes on fitness is even more apparent when shown alongside the corresponding fitness graph (shown on the bottom of Figure 8). The large swing in damping co-efficient at generation 7 corresponds to a matching significant rise in fitness. Other, smaller, fitness gains also appear to have corresponding material value changes.

### Scaling Mesh Resolution

Our second analysis is of the benefits offered by scaling mesh resolution over the course of evolution. Recall that the number of tetrahedra in a mesh are the determining factor in simulation run time, as well as in simulator fidelity. We ran a suite of experiments exploring the effects of different scaling schemes, as summarized in Table 1.

Our intuition was that the the bulk of early evolutionary time, which largely consist of the soft robot flailing around – that is attempting to achieve non-zero fitness, could be performed on relatively low meshes, and then as evolution and fitness progressed, mesh resolution could be scaled upwards to raise the emphasis on fidelity at the cost of longer evaluation times. A similar approach, with rigid-bodied multi-resolution robots, is discussed by Auerbach and Bongard (2010).

There were five mesh resolutions available to the system: low, medium low, med, high, and maximum. Runs could switch 0,1,2 or 4 times. All of the non-static runs shown began on the low mesh – runs that are listed with a mesh switch count of one, for instance, changed from the low mesh to their end mesh. Runs listed with a mesh switch count greater than one ran on an intermediate mesh(es) before reaching their end mesh. All other properties, such as population size, remained constant across experiments. Resolution changes occurred every 30 generations.

Fitness	Hours	Mesh Sequence
40.16	34	Low
46.94	42	Low
24.61	72	Max
14.86	25	MedLow
21.47	25	Low,MedLow
11.97	24	Med
12.86	24	Low,Med
15.99	48	Med
16.65	48	Low,Med
15.91	48	Low,MedLow,Med
6.18	46	Low,Medlow,Med,MedHi,Max

Table 1: A summary of results from resolution scaling

Figure 7, which compares evolution with a single (low-to-medium-low) switch to that with a fixed (medium-low)

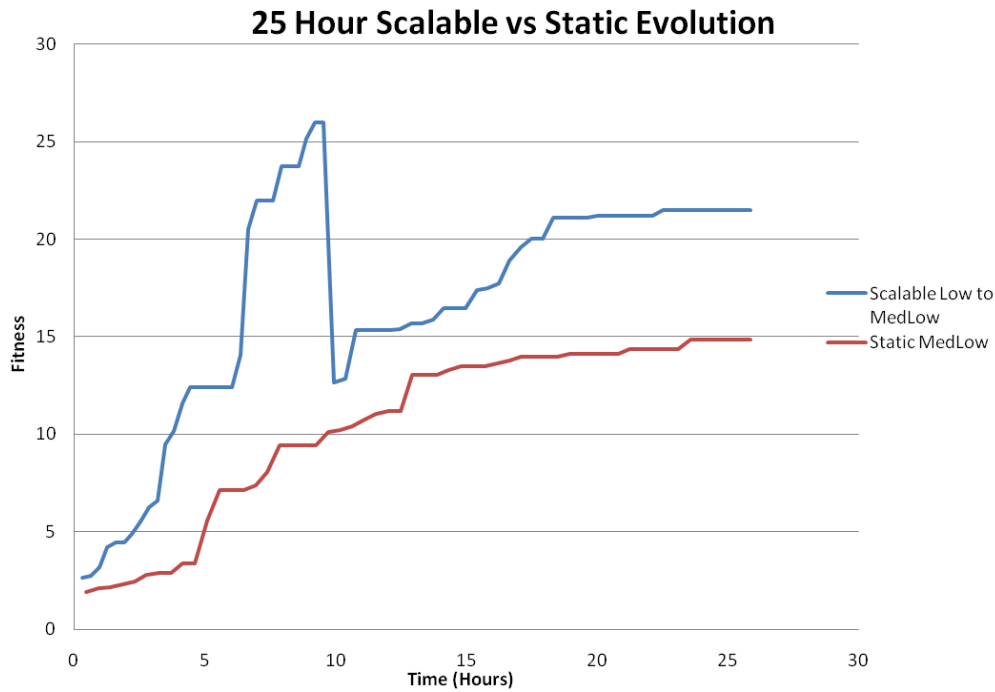


Figure 7: A comparison of co-evolutionary progress on a static mesh resolution (bottom) vs. a single resolution switch. The sudden drop in fitness corresponding to resolution change is caused by the low-resolution gait working less well on the higher resolution mesh.

mesh resolution show the consequences of this process: fitness in the scalable evolution during its “low” phase progresses much more rapidly during the first 10 hours of simulation. Once the phase change into a higher mesh density occurs, however, there is a dramatic drop in fitness, and the scalable run loses much of the ground it had gained (though it still remains above the fixed resolution result). During the following 15 hours, the scalable run is able to make up much of the lost fitness, and improves more rapidly than the static mesh.

This steep loss in fitness is due to the large extent to which the success of a gait is highly tuned to its specific mesh resolution. The same actually holds true of the evolving material properties as well. Gaits and physical properties evolved at one mesh resolution simply do not translate perfectly when placed in a higher resolution simulation.

This dependence on mesh resolution also has a clear effect upon the the maximum obtainable evolutionary fitness: over similar time scales, even the static meshes show significant differences in final fitness.

The last entry in Table 1 illustrates the cost of switching clearly: the final fitness is less than half of that achieved by any other run. This suggests that, in its current form, sometimes the cost of resolution scaling can be too high. Figure 9 shows a case where even a single switch in resolution results in an equivocal, at best, improvement in overall fitness.

The source of this loss in fitness can possibly be illustrated with an interesting qualitative distinction of gaits evolved at varying resolutions: gaits evolved in a low resolution mesh tended to produce bi-pedal gaits, whereas gaits produced in the “maximum” mesh tended to be more bilaterally symmetric, involving instead a forward-propagating wave-like motion. In other words, sauce for the (low-resolution) goose may not be sauce for the (high-resolution) gander. A high-fitness bipedal gait evolved a low mesh resolution ceases to be competitive when placed in a higher-resolution body.

Mesh scaling certainly holds promise, and in a few cases illustrated above, offers an improvement over static-resolution evolution, despite the large fitness drops associated with resolution switches. While it remains to be seen if this is a viable way to address the issue of long simulation times, we are hopeful of its prospects. In future work, we hope to develop a method which allows for more smooth transitions between mesh resolutions.

## Conclusions

In this work we have illustrated the tight coupling which exists between soft body gaits and the body’s underlying material properties. We have also demonstrated how material properties can be fine-tuned to a gait (and vice versa) via co-evolution. Finally, we have explored a method of reducing simulation time by scaling soft body mesh resolution over

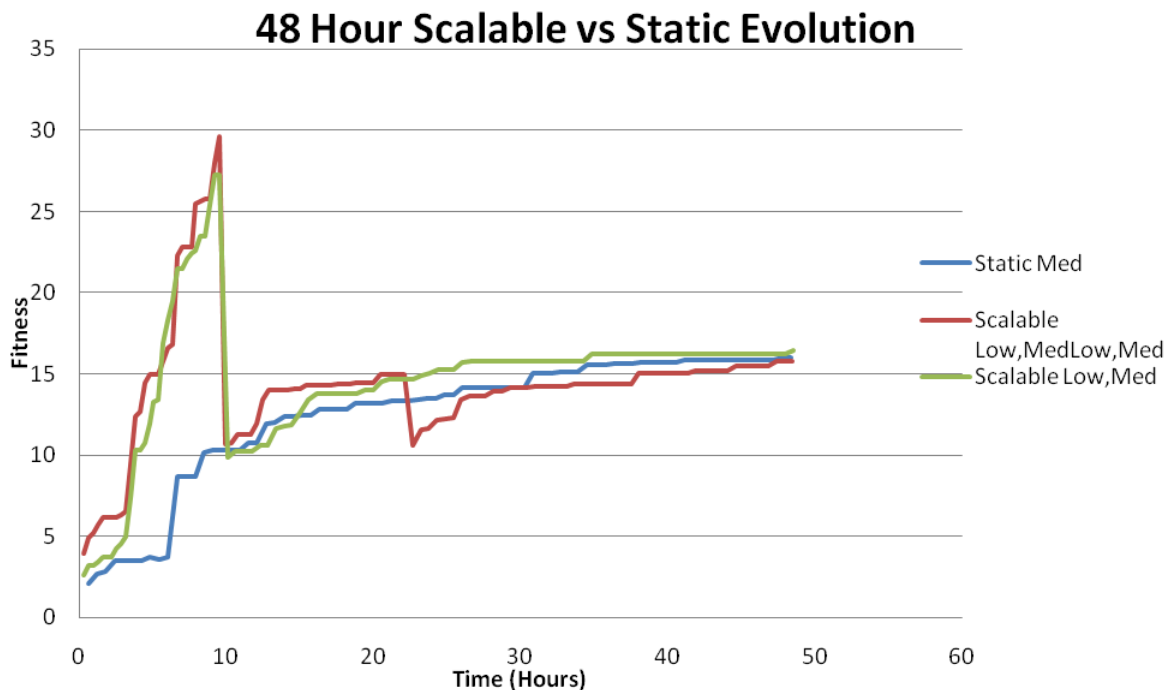


Figure 9: Results comparing the consequence of switching mesh resolution multiple times over the course of evolution. Each time resolution changes there is a dramatic drop in fitness due to the relatively poor translation of gaits and material properties into the higher mesh.

the course of evolution. Collectively, this approach holds promise as a way to discover gaits for dynamically complex soft robots.

This work leaves several compelling unanswered questions which we look forward to addressing. Among them: the mechanisms behind the coupling of material properties and evolved gaits; a comparison to encoding both gait and material properties in a single monolithic genome within a one-population system; an analysis of material property evolutionary trends across resolutions (e.g. a trend toward stiffer bodies as robot speed increases).

Ultimately we are interested in the *trifecta* of simultaneously evolving material properties, gaits, and large-scale morphology (S. Smith and Rieffel, 2010).

In concluding, it is worth emphasizing that this work, while performed in simulation, is grounded in real-world applications. Many ongoing efforts at developing physical soft robots employ silicone elastomers, whose material properties can be changed quite significantly during the mixing process (Trimmer, 2007). Furthermore, this bio-inspired research also ties back into understandings of the incredibly sophisticated biomechanics of completely soft organisms, such as the *Manduca Sexta* caterpillar (Simon et al., 2010).

## References

- Al-Sakran, S. H., Koza, J. R., and Jones, L. W. (2005). Automated re-invention of a previously patented optical lens system using genetic programming. In Keijzer, M., Tettamanzi, A., Collet, P., van Hemert, J. I., and Tomassini, M., editors, *Proceedings of the 8th European Conference on Genetic Programming*, volume 3447 of *Lecture Notes in Computer Science*, pages 25–37, Lausanne, Switzerland. Springer.
- Auerbach, J. and Bongard, J. (2010). Dynamic resolution in the co-evolution of morphology and control. In *Proceedings of the 12th International Conference on the Synthesis and Simulation of Living Systems (ALife XII)*.
- Banzhaf, W. and Harding, S. (2009). Accelerating evolutionary computation with graphics processing units. In *GECCO '09: Proceedings of the 11th annual conference companion on Genetic and evolutionary computation conference*, pages 3237–3286, New York, NY, USA. ACM.
- Lohn, J. D., Hornby, G. S., and Linden, D. S. (2005). An Evolved Antenna for Deployment on NASA's Space Technology 5 Mission. In O'Reilly, U.-M., Riolo, R. L., Yu, T., and Worzel, B., editors, *Genetic Programming Theory and Practice II*. Kluwer.
- Pollack, J., Lipson, H., Funes, P., Ficici, S., and Hornby, G. (1999). Coevolutionary robotics. In *EH '99: Proceedings of the 1st NASA/DOD workshop on Evolvable Hardware*, page 208, Washington, DC, USA. IEEE Computer Society.



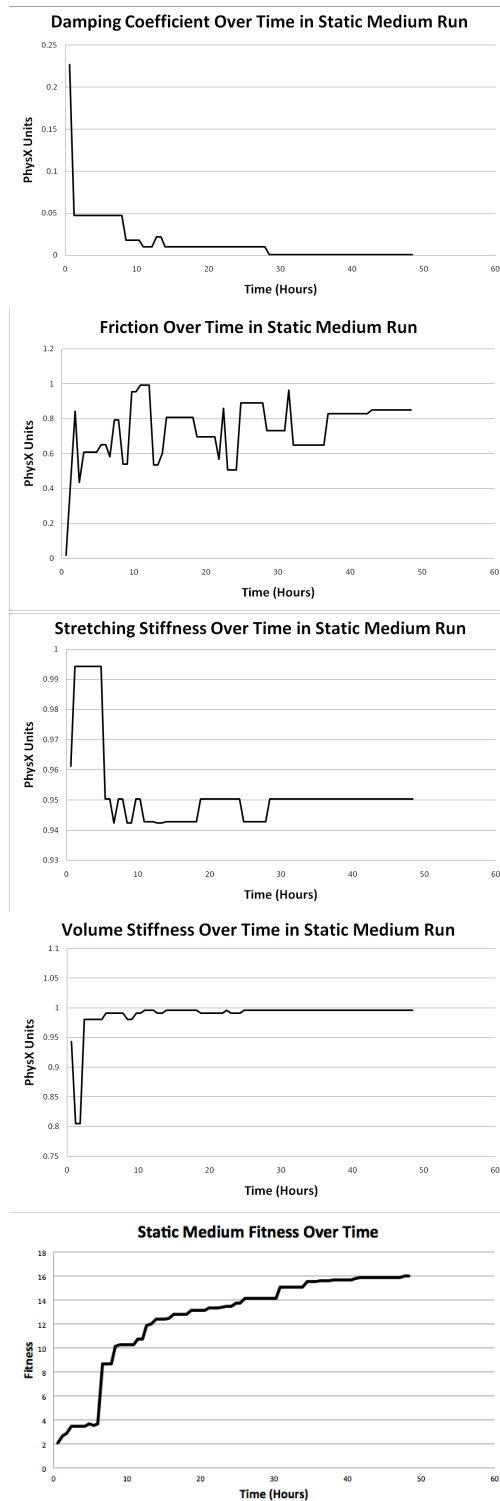


Figure 8: Material property values for population bests per generation. Large swings in values, such as the drop in damping at generation 7, are co-ordinated with large improvements in fitness (bottom figure)

- Pollack, J. B., Lipson, H., Hornby, G., and Funes, P. (2001). Three generations of automatically designed robots. *Artificial Life*, 7(3):215–223.
- Rieffel, J. (2010). Body/brain co-evolution in soft robots (extended abstract). In *Proceedings of the 12th International Conference on the Synthesis and Simulation of Living Systems (ALIFE12)*, pages 257–260. MIT Press.
- Rieffel, J., Saunders, F., Nadimpalli, S., Zhou, H., Hassoun, S., Rife, J., and Trimmer, B. (2009a). Evolving soft robotic locomotion in physx. In *Proceedings of the 2009 Genetic and Evolutionary Computation Conference (GECCO)*.
- Rieffel, J., Valero-Cuevas, F., and Lipson, H. (2009b). Automated discovery and optimization of large irregular tensegrity structures. *Computers & Structures*, 87(5-6):368 – 379.
- S. Smith, S. and Rieffel, J. (2010). A face-encoding grammar for the generation of tetrahedral-mesh soft bodies. In *Proceedings of the 12th International Conference on the Synthesis and Simulation of Living Systems (ALIFE12)*, pages 414–420. MIT Press.
- Simon, M., Woods, W., Serebrenik, Y., Simon, S., Griethuijsen, L. V., Socha, J., Lee, W., and Trimmer, B. (2010). Visceral-locomotory pistoning in crawling caterpillars. *Current Biology*, 20:1458–1463.
- Trimmer, B. (2007). New challenges in biorobotics: incorporating soft tissue into control systems. In *IEEE International Conference on Robotics and Automation*.

# Prediction and Modularity in Dynamical Systems

Artemy Kolchinsky<sup>1,2</sup> and Luis M. Rocha<sup>1,2</sup>

<sup>1</sup> School of Informatics and Computing, Indiana University, Bloomington IN 47401, USA

<sup>2</sup> FLAD Computational Biology Collaboratorium, Instituto Gulbenkian de Ciência, Portugal  
{akolchin,rocha}@indiana.edu

## Abstract

Identifying and understanding modular organizations is centrally important in the study of complex systems. Several approaches to this problem have been advanced, many framed in information-theoretic terms. Our treatment starts from the complementary point of view of statistical modeling and prediction of dynamical systems. It is known that for finite amounts of training data, simpler models can have greater predictive power than more complex ones. We use the trade-off between model simplicity and predictive accuracy to generate optimal multiscale decompositions of dynamical networks into weakly-coupled, simple modules. State-dependent and causal versions of our method are also proposed.

## Introduction

The study of complex dynamical systems – such as gene regulatory networks (Han et al., 2004), structural and functional brain networks (Bullmore and Sporns, 2009), ecological food webs (Krause et al., 2003), and others (Hartwell et al., 1999, Schlosser and Wagner, 2004) – has frequently uncovered the presence of modularity. Broadly speaking, modular systems are composed of tightly-integrated subsystems, called *modules*, which are in turn weakly coupled to one another.

Numerous explanations have been proposed for the function of modularity in complex systems, only a few of which are mentioned here. Simon (1962) suggested that modularity can contain the effects of harmful perturbations and lead to greater developmental and operational robustness, especially when modules are hierarchically arranged. Kashtan and Alon (2005) argued that modular systems can take advantage of reusability when adapting to changing combinations of fixed environmental tasks. Tononi et al. (1998) proposed that modularity balances the conflicting needs for subsystems that are functionally specialized but also integrated into globally coherent states. Notably, it has also been shown to arise as a result of non-adaptive processes, such as neutral evolution of gene regulatory networks (Force et al., 2005, Solé and Valverde, 2008) and stochastic fluctuations in network connectivity patterns (Guimera et al., 2004).

Though the concept of modularity has acquired a central place in the study of complex systems, its meaning and operationalization varies widely between scientific paradigms, fields, and processes of interest. In the biological sciences alone, one can find references to *structural*, *developmental*, *physiological*, *variational*, and *functional modularity* (Winther, 2001, Wagner et al., 2007), among others. In this work, we propose a formal notion of modularity based on statistical modeling. Our approach applies to a broad class of discrete-time multivariate dynamics, whether represented by dynamic models, such as Boolean or dynamic Bayesian networks, or empirical distributions estimated from time series recordings. Unlike much recent work on community-structure in static graphs, we identify modularity in the organization of dynamically interacting components. We argue that in addition to being useful for analysis of real-life dynamical systems, our approach can shed light on connections between notions of modularity utilized in different domains, as well as the general role of modularity in modeling.

The next section provides a brief background on information theory. We then outline traditional information-theoretic approaches to modularity in dynamical systems, and develop our own treatment in terms of statistical modeling. After applying it to an example dynamical system, we consider state-dependent and causal versions of modular decompositions. We conclude by discussing issues of parameterization, directions for further work, and connections between our method and broader questions of modeling.

## Information theory

Information theory provides principled measures of information transfer and statistical dependence in distributed systems. As such, it is well-suited for quantifying measures of coupling and modularity.

To review, Shannon *entropy* measures the uncertainty in the measurement outcomes of a random variable. If  $X$  is a discrete random variable with an associated probability distribution  $P(X)$ , then its entropy is:

$$H(X) = - \sum_{x \in X} P(x) \log P(x)$$



A random variable that takes a single value with probability 1 has an entropy of 0, while an equiprobable random variable assumes the maximum entropy of  $\log |X|$ , where  $|X|$  is the number of possible outcomes. When the base of the logarithm is 2, as in this work, the units of entropy are *bits* (1 bit is the uncertainty in the choice between 2 equally possible outcomes). Because measuring a variable reduces uncertainty about its value, entropy can also be considered a measure of information.

When provided with a joint distribution over two random variables such as  $P(X, Y)$ , *conditional entropy* measures the expected uncertainty in the value of one variable given that the value of the other is known:

$$H(X|Y) = H(X, Y) - H(Y) = - \sum_{x,y} P(x, y) \log P(x|y)$$

*Mutual information* is a symmetric measure of nonlinear correlation between two random variables. Expressed as the difference between entropy and conditional entropy, it can be interpreted as the reduction in uncertainty about the value of one random variable provided by knowledge of the other:

$$\begin{aligned} I(X; Y) &= H(X) + H(Y) - H(X, Y) \\ &= H(X) - H(X|Y) = H(Y) - H(Y|X) \\ &= \sum_{x,y} P(x, y) \log \frac{P(x, y)}{P(x)P(y)} \end{aligned}$$

Mutual information captures the amount of constraint in the joint distribution of two variables not present in their marginal distributions. It is equal to 0 when two variables are statistically independent, and reaches its maximum possible value of  $\min\{H(X), H(Y)\}$  when one variable is a function of the other.

Mutual information can be extended to the case of more than two variables. Let random vector  $\mathbf{X} = (X_1, X_2, \dots, X_L)$  with distribution  $P(\mathbf{X})$  represent the state of a system composed of  $L$  distinct variables. The total constraint in this system not present in any single variable is measured by a multivariate version of mutual information, often called *multi-information* (Studený and Vejnarova, 1998) or *integration* (Tononi et al., 1994):

$$\begin{aligned} \mathcal{I}(\mathbf{X}) &= \sum_{i=1}^L H(X_i) - H(\mathbf{X}) \\ &= \sum_{\mathbf{x}} P(\mathbf{x}) \log \frac{P(\mathbf{x})}{\prod_{i=1}^L P(x_i)} \end{aligned} \quad (1)$$

*Kullback-Leibler (KL) divergence* is a measure of the difference between two distributions:

$$KL(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)} \quad (2)$$

It is always positive and 0 iff  $P = Q$ , though it is not a

distance because it is not symmetric. Importantly, many information-theoretic measures can be restated in terms of KL divergence. For example, the multi-information of eq. 1 is equal to the KL divergence between the distribution of  $\mathbf{X}$  and a product of the marginal distributions over the individual variables of  $\mathbf{X}$ .

## Modularity in multivariate dynamics

As previously mentioned, multi-information measures the total amount of higher-order constraint present among the variables of a multivariate system. It is 0 when these variables are independent, and increases when more statistical interaction between variables is present (Studený and Vejnarova, 1998). For this reason, many formal approaches to modularity search for system transformations that minimize this measure.

Several kinds of transformations can be investigated. *Independent component analysis* attempts to minimize multi-information over the space of linear mappings (coordinate changes) of a multivariate system (Hyvärinen and Oja, 2000). A different approach, closer to the one pursued here, looks for *partitions* of system variables with low multi-information.

A partition  $\pi$  of set  $S$  is a set of mutually exclusive, nonempty subsets  $B \subseteq S$ , called *blocks*, such that  $\bigcup_{B \in \pi} B = S$ . For example,  $\{\{1\}, \{2, 3\}\}$  and  $\{\{1, 2, 3\}\}$  are two possible partitions of the set  $\{1, 2, 3\}$ . We also use a more concise notation: the two partitions above, for example, can be referred to as 1/23 and 123 respectively. Additionally,  $\pi_0$  is used to indicate the *total partition*, which includes the entire set in a single block, i.e.  $\pi_0 \equiv \{S\}$ .

We look at partitions of  $V = \{1, \dots, L\}$ , the set of indexes of the variables of random vector  $\mathbf{X}$ . For partition  $\pi$  and block  $B \in \pi$ ,  $P(\mathbf{X}_B)$  indicates the marginalization of  $P(\mathbf{X})$  onto the variables whose indexes are in  $B$ . For example,  $P(\mathbf{X}_{\{1,2\}})$  is the marginal distribution of the first two variables of  $\mathbf{X}$ .

We define the multi-information of partition  $\pi$  as:

$$\mathcal{I}_\pi(\mathbf{X}) = \sum_{B \in \pi} H(\mathbf{X}_B) - H(\mathbf{X})$$

This measure quantifies the amount of constraint holding among the blocks of  $\pi$ . Finding partitions with low multi-information corresponds to identifying weakly-coupled subsystems. Variations on this theme appear in information-theoretic treatments of modularity starting from early cybernetics (Conant, 1972) to more recent approaches in computational neuroscience (Tononi and Sporns, 2003).

Multi-information is defined over a time-invariant distribution of system states. Though it does not account for the dynamic flow of information within a system, it can be generalized to this case. Assume a multivariate system with Markovian dynamics represented by  $P(\mathbf{X}' = \mathbf{x}' | \mathbf{X} = \mathbf{x})$ , the conditional probability distribution of transitioning to

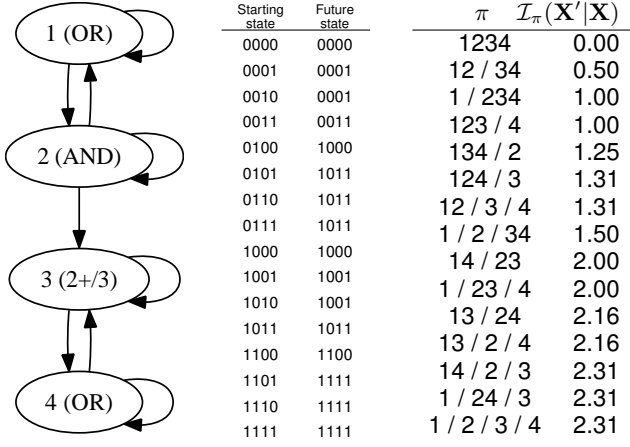


Figure 1: A simple four node Boolean network (nodes 1, 2, 3, and 4 perform OR, AND, majority, and OR update functions respectively). Its full state transition table is shown in center. On the right, the stochastic interaction of every possible partition of the network.

each future state  $\mathbf{x}'$  given starting state  $\mathbf{x}$ , as well as  $P(\mathbf{X} = \mathbf{x})$ , the distribution over starting states.<sup>1</sup> The amount of information flowing dynamically among the blocks of  $\pi$  is called *stochastic interaction* (Ay and Wennekers, 2003). It is a conditional version of KL divergence between the transition distribution of the whole system and the product of marginal transition distributions of the variable blocks specified by partition  $\pi$ :

$$\begin{aligned} \mathcal{I}_\pi(\mathbf{X}'|\mathbf{X}) &= \sum_{B \in \pi} H(\mathbf{X}'_B|\mathbf{X}_B) - H(\mathbf{X}'|\mathbf{X}) \quad (3) \\ &= \text{KL} \left[ P(\mathbf{X}'|\mathbf{X}) \left\| \prod_{B \in \pi} P(\mathbf{X}'_B|\mathbf{X}_B) \right\| \right] \end{aligned}$$

These kinds of dynamic generalizations of multi-information have recently been proposed as measures of system-wide coupling in brain dynamics (Balduzzi and Tononi, 2008, Barrett et al., 2011).

A simple demonstration is provided by the Boolean network in fig. 1. It has four nodes, whose update functions are OR, AND, majority rule, and OR respectively. The stochastic interaction of each possible partition is provided, assuming a uniform distribution over starting states. For example, the partition 12/34 is the bi-partition having the lowest stochastic interaction: the block  $\{1, 2\}$  has conditional entropy  $H(\mathbf{X}'_{\{1,2\}}|\mathbf{X}_{\{1,2\}}) = 0$  (nodes 1 and 2 do not depend on the rest of the system, so their marginalized dynamics are deterministic), while block  $\{3, 4\}$  has conditional entropy  $H(\mathbf{X}'_{\{3,4\}}|\mathbf{X}_{\{3,4\}}) = 0.5$ . Because the system as a

<sup>1</sup>We assume that the dynamics are stationary, in that the transition probability distribution does not change through time. Our analysis can also be applied to higher-order Markovian systems, though for simplicity they are not considered here.

whole is deterministic,  $H(\mathbf{X}'|\mathbf{X}) = 0$  and the total stochastic interaction of partition 12/34 is  $H(\mathbf{X}'_{\{1,2\}}|\mathbf{X}_{\{1,2\}}) + H(\mathbf{X}'_{\{3,4\}}|\mathbf{X}_{\{3,4\}}) - H(\mathbf{X}'|\mathbf{X}) = 0.5$ .

Unfortunately, stochastic interaction is not a suitable cost function for identifying modular partitions of a multivariate dynamical system (similarly for multi-information and multivariate non-dynamical systems). In any such system, a minimal stochastic interaction of 0 will be assigned to the total partition  $\pi_0$ , and generally a partition will never have a greater stochastic interaction than any of its refinements (where one partition is a *refinement* of another if every block of the former is a subset of some block of the latter). Selecting partitions using stochastic interaction will thus favor partitions with large blocks, the total partition being a (possibly non-unique) global minimum.

Due to this, several authors have proposed normalizing factors that penalize large partitions (Conant, 1972, Balduzzi and Tononi, 2008). However, the derivation and justification of these normalizing terms is ad hoc. In this work, we approach the problem of identifying modules from the point of view of statistical prediction. This yields principled penalization terms for large partitions and leads us to uncover modular decompositions with clear interpretations in terms of statistical modeling.

## Statistical modeling and modular decompositions

Information theory is intimately connected with statistical modeling (Rissanen, 2007). For example, assume a model that assigns a probability value to data  $\mathbf{x}$ :

$$Q(\mathbf{x}) = \int_{\Theta} Q(\mathbf{x}|\theta) \omega(\theta) d\theta \quad (4)$$

This term, called the *marginal likelihood* in the Bayesian literature, is the expectation of the likelihood function  $Q(\mathbf{x}|\theta)$  with respect to distribution  $\omega(\theta)$  over parameter values.

$Q(\mathbf{x})$  is a measure of predictive fit to data, and its logarithm is often maximized over parameter distributions or model choices. Equivalently, one can minimize the negative of its logarithm, a measure of predictive error called *log loss*. If data samples are drawn from some true probability distribution  $P(\mathbf{X} = \mathbf{x})$ , then the expectation of the log loss of the marginal likelihood is:

$$-\sum_{\mathbf{x} \in \mathbf{X}} P(\mathbf{x}) \log Q(\mathbf{x}) = \text{KL}(P\|Q) + H(P(\mathbf{X}))$$

The KL term (from eq. 2) is non-negative, and reaches its minimum of 0 when the model is perfectly fit, i.e.  $Q = P$ . It is a measure of excess prediction error of the model above the minimum possible. This minimum is specified by the entropy term, and depends only on the true distribution  $P(\mathbf{X})$  and not on model or parameter choices.

A similar situation holds in the dynamic setting. We call

*dynamic models* those that generate conditional distributions of multivariate future states  $\mathbf{x}'$  given starting states  $\mathbf{x}$ :

$$Q(\mathbf{x}'|\mathbf{x}) = \int_{\Theta} Q(\mathbf{x}'|\mathbf{x}, \theta) \omega(\theta) d\theta$$

We look at statistical prediction of dynamical systems from the perspective of an agent who does not possess a perfectly fit model, but must learn a dynamic model given previous observations. The agent is provided with a set of factorized models: for each partition of system variables  $\pi$ , there is a dynamic model  $Q_{\pi}$  whose parameters and marginal likelihood obey the independence conditions imposed by the block structure of  $\pi$ :

$$Q_{\pi}(\mathbf{x}'|\mathbf{x}) = \prod_{B \in \pi} Q_{\pi}(\mathbf{x}'_B|\mathbf{x}_B) \quad (5)$$

The predictive performance of our agent depends on the chosen model and the amount of previously observed data. It can be quantified with a *risk function*, which here is the KL divergence between the true distribution  $P(\mathbf{X}'|\mathbf{X})$  and the distribution predicted by a dynamic model (Haussler and Oppel, 1997). The risk of model  $Q_{\pi}$  on the next sample, after observing  $N$  previous samples, is:

$$r_{N, Q_{\pi}} = \text{KL}[P(\mathbf{X}'|\mathbf{X}) \| Q_{\pi}(\mathbf{X}'|\mathbf{X}, \mathbf{X}^{1..N}, \mathbf{X}^{1..N})] \quad (6)$$

The expectation in the KL term is taken over the next sample of  $\mathbf{X}'$ ,  $\mathbf{X}$ , as well as  $N$  previous i.i.d. samples  $\mathbf{X}^{1..N}$ ,  $\mathbf{X}^{1..N}$ . The Bayesian *posterior predictive distribution*:

$$Q_{\pi}(\mathbf{x}'|\mathbf{x}, \mathbf{x}^{1..N}, \mathbf{x}^{1..N}) = \int Q_{\pi}(\mathbf{x}'|\mathbf{x}, \theta) Q_{\pi}(\theta|\mathbf{x}^{1..N}, \mathbf{x}^{1..N}) d\theta$$

is the marginal likelihood of eq. 4, with the distribution over parameter values conditioned on  $N$  previous data samples. From the point of view of machine learning, such Bayesian updating of parameters in light of observed data corresponds to model *training*, while evaluating the expected model risk on new samples corresponds to model *testing*. More concretely, our dynamic models can be considered *supervised learners*: given data, they infer probabilistic mappings from inputs (starting states  $\mathbf{X}$ ) to outputs (future states  $\mathbf{X}'$ ).

Given the independence assumption of eq. 5, risk  $r_{N, Q_{\pi}}$  becomes:

$$\mathcal{I}_{\pi}(\mathbf{X}'|\mathbf{X}) + \sum_{B \in \pi} \text{KL}[P(\mathbf{X}'_B|\mathbf{X}_B) \| Q_{\pi}(\mathbf{X}'_B|\mathbf{X}_B, \mathbf{X}_B^{1..N}, \mathbf{X}_B^{1..N})]$$

This form draws attention to the two components that contribute to risk (that is, predictive error). The *stochastic interaction term* (see also eq. 3) arises as a consequence of ignoring dynamic coupling between variables in different blocks. It is the minimal excess error of a factorized model (in which the dynamics of the variable blocks induced by partition  $\pi$  are independent) above an optimally fit whole-system model (where interactions between all variables can be captured).

The second term, called the *complexity term*, reflects the excess predictive error of a trained model above the minimum possible. It arises because a model trained on a finite amount of data maintains some uncertainty about optimal parameter values. For a given amount of training data, complex models (with larger parameter spaces) will have greater parameter uncertainty than simpler models, resulting in more excess predictive error. As  $N \rightarrow \infty$ , the complexity term can be asymptotically approximated by  $\frac{d_{\pi}}{2N}$ , where  $d_{\pi}$  refers to the number of parameters of model  $Q_{\pi}$  (Komaki, 1996, Barron and Hengartner, 1998). This yields:<sup>2</sup>

$$r_{N, Q_{\pi}} \approx \mathcal{I}_{\pi}(\mathbf{X}'|\mathbf{X}) + \frac{d_{\pi}}{2N} \quad (7)$$

For a given amount of training data  $N$ , the model with the lowest risk,

$$Q^*(N) = \arg \min_{Q_{\pi}} r_{N, Q_{\pi}}$$

corresponds to the partition providing an optimal predictive decomposition of the system. Models that minimize risk offer a balance between two conflicting constraints: on one hand, low stochastic interaction (better predictions under optimal fit), on the other, low model complexity (easier parameter estimation with limited training data). Because partitions with smaller blocks (which have smaller-state-space dynamics representable by fewer parameters) generally induce simpler models, risk presents a principled cost function for identifying small, weakly-coupled modules. The amount of data  $N$  parameterizes this trade-off: as  $N$  increases, emphasis is shifted from the complexity term to the stochastic interaction term, and groups of variables whose dynamic interactions carry the most information while being easiest to learn are first to coalesce into multivariate blocks of the optimal model.<sup>3</sup> Thus, selecting optimal decompositions while increasing the amount of training data generates a modular multiscale decomposition of system variables. In the infinite data limit, the risk of each model  $Q_{\pi}$  reaches its minimum of  $\mathcal{I}_{\pi}(\mathbf{X}'|\mathbf{X})$ , and the partition corresponding to  $Q^*$  becomes the one with lowest stochastic integration (the total partition being a possibly non-unique minimum).

## Decomposing a dynamical system

The complexity term in eq. 7 depends on the parametric form of the dynamic model. Though a variety of possibili-

<sup>2</sup>This approximation assumes continuously-parameterized models and standard regularity conditions. It also assumes that, for all  $\pi$ , some parameterization of  $Q_{\pi}$  offers a perfect fit to the factorized  $\prod_{B \in \pi} P(\mathbf{X}'_B|\mathbf{X}_B)$ . It is possible to generalize beyond this case, where the factorizations of the true distribution are 'out-of-class' of the models  $Q_{\pi}$ .

<sup>3</sup>Minimizing risk can be seen as a form of information bottleneck (Tishby et al., 1999): it searches for factorized models whose parameters minimize information about training data while maximizing information about system dynamics; the size of the training data serves as a trade-off parameter.

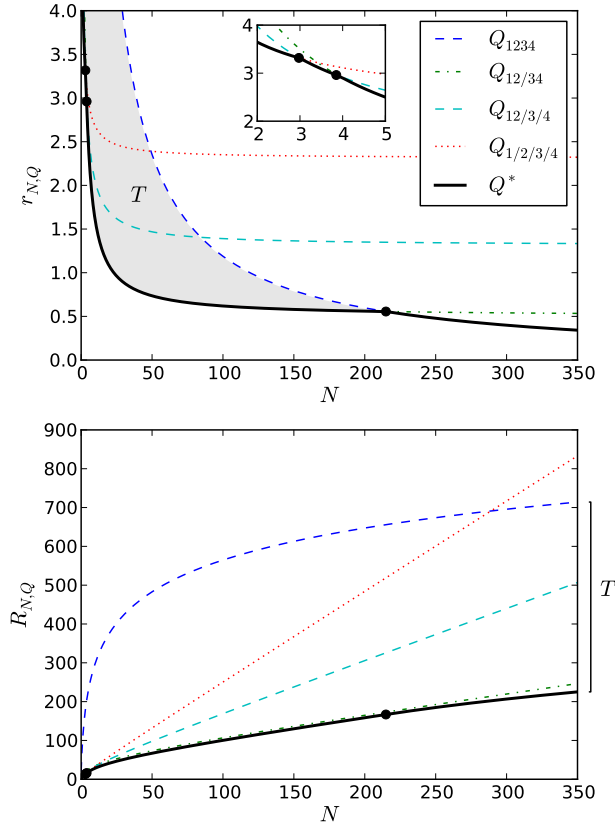


Figure 2: Top: approximate risk for optimally-predictive models of the Boolean network from fig. 1. Dots mark switches of the optimal model  $Q^*$ ; inset shows first two switches. Bottom: cumulative risk, or total accumulated prediction error for models plotted in the top graph. Total modularity ( $T$ ) is asymptotic difference between cumulative risks of  $Q_{1234}$  and  $Q^*$  or, alternatively, area between lines corresponding to (non-cumulative) risks of  $Q_{1234}$  and  $Q^*$ .

ties exist, here our dynamic models are assumed to be products of first-order Markov chains with Dirichlet priors. The number of parameters of model  $Q_\pi$  from this class is:

$$d_\pi = \sum_{B \in \pi} |\mathbf{X}_B| (|\mathbf{X}'_B| - 1) \quad (8)$$

where  $|\mathbf{X}_B|$  is the number of supported starting state outcomes and  $|\mathbf{X}'_B|$  is the number of possible future state outcomes of the variables with indexes in block  $B$ . For example, for a single block of Boolean variables with a fully supported starting state distribution, these are both equal to  $2^{|B|}$ . For this model class, the complexity term scales exponentially with the number of variables in each block.

As an example, we look at optimal decompositions of the network in fig. 1. Its risk, calculated using the approximation of eq. 7 and parameter counts of eq. 8, is shown at the

top of fig. 2.<sup>4</sup> The risk is plotted for those models which reach minimum risk at some point of the training process, as well as that of the overall minimal risk model  $Q^*$  at each  $N$ . Predictive power is initially optimized by the model corresponding to partition 1/2/3/4 (the simplest model which treats all nodes independently). At  $N \approx 3$  (inset), it is replaced by the model corresponding to partition 12/3/4 (variables 1 and 2 now merged into a single block); at  $N \approx 4$  (inset), by the model corresponding to partition 12/34; and finally at  $N \approx 215$ , the most predictive model becomes the one corresponding to the total partition 1234.

## Total modularity

So far, our measure of modularity has been parameterized by  $N$ , the amount of training data. Here, we derive a parameter-free measure of the *total modularity* in a dynamical system.

In our definition of risk (eq. 6), we used the *posterior predictive distribution*  $Q_\pi(\mathbf{X}' | \mathbf{X}, \mathbf{X}^{1..N}, \mathbf{X}^{1..N})$ , the probability assigned to the next data sample by a model trained on  $N$  previous data samples. Given our assumptions, the following relationship holds between the *prior predictive distribution*, the probability an untrained model assigns to  $N$  data samples, and the posterior predictive distribution:

$$Q_\pi(\mathbf{X}'^{1..N} | \mathbf{X}^{1..N}) = \prod_{n=0}^{N-1} Q_\pi(\mathbf{X}'^{n+1} | \mathbf{X}^{n+1}, \mathbf{X}^{1..n}, \mathbf{X}^{1..n})$$

This suggests the *prequential* interpretation of Bayesian prediction (Dawid, 1992): the expected predictive error of a model on  $N$  samples is the sum of the expected predictive errors on each successive sample after training on the previous samples. This accumulated prediction error is termed *cumulative risk* (Haussler and Oppner, 1997):

$$R_{N,Q_\pi} = \sum_{n=0}^{N-1} r_{n,Q_\pi}$$

The risk of eq. 6 can be seen as the rate of change of the cumulative risk as the amount of training data grows.

*Total modularity* is the total gain in predictive accuracy (i.e., decrease in cumulative risk) provided by the optimally predictive models  $Q^*(N)$  versus the unfactorized, total-partition model  $Q_{\pi_0}$ . Let  $R_{N,Q^*} = \sum_{n=0}^{N-1} r_{n,Q^*(n)}$  be the cumulative risk of an agent who selects the risk-minimal model at each  $N$ . The total modularity is then:

$$T = \lim_{N \rightarrow \infty} (R_{N,Q_{\pi_0}} - R_{N,Q^*}) \quad (9)$$

Total modularity measures the overall predictive advantage gained by using factorized models, and is not a function of a particular  $N$ . High values of total modularity indicate

<sup>4</sup>In general, the approximation of eq. 7 is only accurate for large  $N$ . However, it suffices for our explanatory purposes.

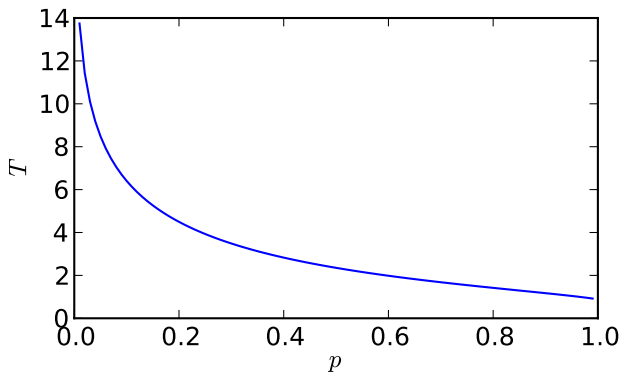


Figure 3: Total modularity of two binary variables which copy each others' state with probability  $p$  and maintain their own state with probability  $1 - p$ . Total modularity increases as coupling decreases, and diverges as  $p \rightarrow 0$ .

that simpler models have significantly improved predictive performance during earlier stages of the learning process.<sup>5</sup> To use the previous example, the cumulative risk of the models plotted at the top of fig. 2 is shown at the bottom of that figure. The total modularity of the dynamic network shown in fig. 1 is equal to the asymptotic difference between the cumulative risks of  $Q_{1234}$  ( $= Q_{\pi_0}$ ) and  $Q^*$ . Equivalently, it is also the total area between the lines corresponding to the (non-cumulative) risks of  $Q_{1234}$  and  $Q^*$ .

For another illustration of total modularity, we consider a simple dynamical system composed of two binary variables. Each variable is parameterized in the following manner: at each time step, with probability  $p$  it assumes the value of the other variable in the previous time step, and with probability  $1 - p$  it maintains its own value from the previous time step. The amount of dynamic coupling between the two nodes increases with  $p$ : at  $p = 0$  the variables have no interaction, while at  $p = 1$  their values are completely correlated (with a one timestep lag). This dynamic coupling is illustrated in fig. 3, which plots the total modularity of this system against the coupling parameter  $p$ . The total modularity monotonically decreases as  $p$  increases, showing that greater coupling leads to lower total modularity. As  $p \rightarrow 0$ , the two variables become completely independent and total modularity diverges (in this case, it grows without bound at a rate proportional to  $\log N$ ).

### State-dependent and causal modularity

The way information flows within a dynamical system can depend on the system's state. For example, a partition's stochastic interaction can be different in different attractors. We can quantify this by different choices of the starting

<sup>5</sup>Minimization of accumulated error by online switching from simpler to more complex models is related to a learning framework recently proposed by van Erven et al. (2007)

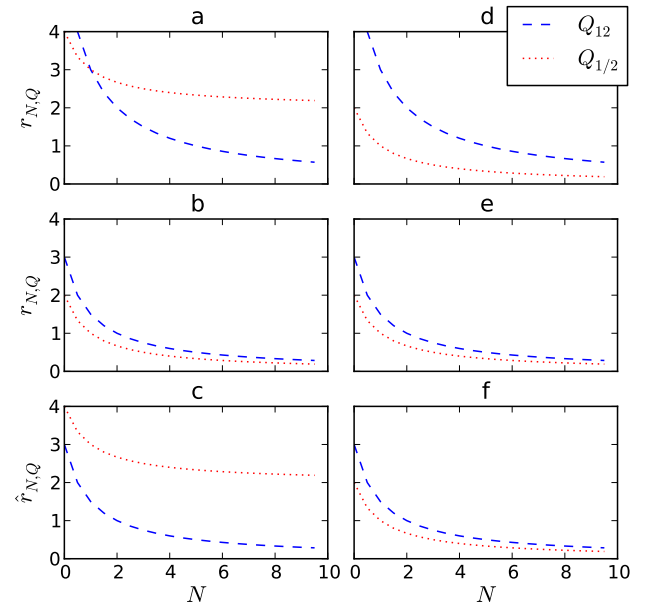


Figure 4: Risk for two systems, each having two binary variables: in system  $A$  (left column) each variable copies previous value of the other, in system  $B$  (right column) each variable takes opposite of its own previous state. a) and d): Risk under uniform starting state distribution. Lowest risk model of  $A$  becomes the total one, while factorized model remains optimal for  $B$ . b) and e): Risk and optimal decompositions depend on the starting state distribution. Computed over  $P(\mathbf{X} = (0, 1)) = 0.5, P(\mathbf{X} = (1, 0)) = 0.5$ , risk and optimal decompositions become the same for  $A$  and  $B$ , though their causal organization is different. c) and f): Causal risk leads to different decompositions of  $A$  and  $B$ , even when computed over same starting state distribution as in b) and e).

state distribution,  $P(\mathbf{X})$ . Though we have generally taken  $P(\mathbf{X})$  to be a fully-supported uniform distribution, it can be weighted preferentially over some subset of starting states.

For example, consider two systems, each composed of two binary variables. In system  $A$ , each variable copies the previous value of the other, while in system  $B$ , each variable takes the opposite of its own previous state. Fig. 4 shows the risk plots for both  $A$  (left column) and  $B$  (right column), where 4a and 4d are calculated for a uniform starting state distribution. The risk, as well as the optimal decompositions, is different between the two systems:  $A$  (which performs the copy operation) eventually chooses the total partition  $\{\{1, 2\}\}$  as the most predictive, while  $B$  (whose variables perform independent state flips) never does.

If, however, a non-uniform starting state distribution is chosen, risk and optimal decompositions can change. The risk for starting state distribution  $P(\mathbf{X} = (0, 1)) = 0.5, P(\mathbf{X} = (1, 0)) = 0.5$  are shown in fig. 4b and 4e (for systems  $A$  and  $B$  respectively). Different parts of the start-

ing state space induce different risk values and optimal decompositions: for this distribution, fig. 4b shows that the total partition  $\{\{1, 2\}\}$  is never chosen as the optimally predictive one for system  $A$ .

Additionally, for these starting states the transition distributions of  $A$  and  $B$  are identical: if either system is started in state  $(0, 1)$ , it deterministically transitions to state  $(1, 0)$ , and similarly for the transition from  $(1, 0)$  to  $(0, 1)$ . Because the observed dynamics of the two systems are identical, the risk functions and optimal decompositions are also equal. Though systems  $A$  and  $B$  are defined using different causal architectures, here their modular organizations are indistinguishable. Specifically,  $A$  is postulated to have a causal connection among its variables but – for this starting state distribution – they display no stochastic interaction.

This example highlights the difference between statistical correlation and causal interaction. To properly handle the latter, we utilize a notion of causality based on semantics of intervention (Pearl, 2000), recently developed in an information-theoretic direction by Ay and Polani (2008). In Pearl’s treatment, conditional probability distributions represent not only correlations, but also responses of variables to externally-imposed interventions. This is especially natural when dynamics of interest are generated by causal models, such as dynamic causal Bayesian or Boolean network models frequently used in artificial life and systems biology.

In our example, the functional organization of systems  $A$  and  $B$  can be differentiated – even within the non-uniform starting state distribution mentioned above – if the starting states of the systems can be intervened upon. This is because in system  $A$  – but not system  $B$  – changing the starting state of one variable can change the other variable’s future state.

We consider interventions formally by noting that the risk  $r_{N, Q_\pi}$  of eq. 6 need not take the same starting state distribution for training data as for the testing data. Instead, we take the starting state distribution for training data to be drawn i.i.d. from a fully-supported and uniform distribution  $\hat{P}(\mathbf{X})$  (the distribution of interventions), while the testing starting states can be drawn from any  $P(\mathbf{X})$  of interest. We refer to risk evaluated under this learning scenario as *causal risk*:

$$\hat{r}_{N, Q_\pi} = \sum_{\mathbf{x}, \mathbf{x}'} P(\mathbf{x}) P(\mathbf{x}' | \mathbf{x}) \left[ \log P(\mathbf{x}' | \mathbf{x}) - \sum_{\mathbf{x}^{1..N}, \mathbf{x}'^{1..N}} \hat{P}(\mathbf{x}^{1..N}) P(\mathbf{x}'^{1..N} | \mathbf{x}^{1..N}) \log Q_\pi(\mathbf{x}' | \mathbf{x}, \mathbf{x}'^{1..N}, \mathbf{x}^{1..N}) \right]$$

As  $N \rightarrow \infty$ , the posterior predictive distribution of model  $Q_\pi$  approaches  $\prod_{B \in \pi} \hat{P}(\mathbf{X}'_B | \mathbf{X}_B)$ , where  $\hat{P}(\mathbf{X}'_B | \mathbf{X}_B)$  is the whole-system transition distribution  $P(\mathbf{X}' | \mathbf{X})$  marginalized onto variables in block  $B$  using  $\hat{P}(\mathbf{X})$ . Then,  $\hat{r}_{N, Q_\pi}$  can be approximated by:

$$\mathcal{I}_\pi(\mathbf{X}' | \mathbf{X}) + \sum_{B \in \pi} \text{KL} \left[ P(\mathbf{X}'_B | \mathbf{X}_B) \parallel \hat{P}(\mathbf{X}'_B | \mathbf{X}_B) \right] + \frac{d_\pi}{2N}$$

where  $\mathcal{I}_\pi$ ,  $d_\pi$ , and the expectations in the KL terms use the testing starting state distribution. The KL divergence between  $P(\mathbf{X}'_B | \mathbf{X}_B)$  (the whole-system transition distribution marginalized onto variables in block  $B$  using  $P(\mathbf{X})$ ) and  $\hat{P}(\mathbf{X}'_B | \mathbf{X}_B)$  reflects the amount of extra perturbation that active interventions inject into block dynamics. The two distributions need not be equal, unless  $P(\mathbf{X}) = \hat{P}(\mathbf{X})$  or the partition under consideration is the total one. Because KL divergence is non-negative, causal risk  $\hat{r}_{N, Q_\pi}$  is not less than the statistical risk  $r_{N, Q_\pi}$  (compare above to eq. 7).

Fig. 4c and 4f show the causal risk for systems  $A$  and  $B$  (respectively) with  $P(\mathbf{X} = (0, 1)) = 0.5$ ,  $P(\mathbf{X} = (1, 0)) = 0.5$ . In 4c – but not 4f – the total partition model assumes a lower risk than the factorized model, indicating that for the starting states in question, system  $A$  – but not system  $B$  – has causal interactions between its variables.

## Conclusion

Modularity is normally treated as an objective property of a system’s organization. Our approach instead considers from the perspective of modeling and prediction. In the context of inferring dynamic models from limited data, modularity allows for models that are predictive but simple, with the amount of training data controlling the trade-off. Our statistical treatment connects to previous information-theoretic approaches, but goes further by providing principled terms for identifying small modules.

Our approach can also be used to find state-dependent modular organizations, both in statistical and causal (interventional) senses: models trained on interventional dynamics but tested on arbitrary distributions give rise to a measure that identifies causal modules. This is related to existing information-theoretic measures of causal interactions between subsystems (Tononi and Sporns, 2003), but here emerges naturally from the framework of statistical modeling. This framework also produces a measure of total modularity present in the system, which quantifies the overall predictive advantage that modularity provides through the entire model inference process.

As a side note, if the learning of real-world cognitive systems (such as scientists or organisms) proceeds in a manner somewhat similar to the statistical framework presented here, our approach suggests why such systems may infer modular organizations in the external world: under conditions of limited data, this assumption can simplify learning and lead to gains in predictive power.

One important issue with our treatment is its model-dependence. The complexity penalization term of eq. 6 depends on the model class, and different model classes may have different parameterizations and functional forms. Our examples employed products of Markov chain models, a rather general dynamic model class but one heavily parameterized; others could be used. The choice of model class can be thought of as a null model of system dynamics.

Several generalizations suggest themselves. For example, it is possible to infer module timescales by searching not only over decompositions, but also model orders (numbers of previous states on which transition probabilities depend; for inferring Markov chain order, see Strelhoff et al., 2007). Fuzzy modular organizations, in which a variable can belong to more than one module, can be accommodated by allowing partially-overlapping blocks. More generally, the model search space could include other structures besides partitions (e.g. trees or networks) to impose independence constraints on information flow between blocks.

Identifying modularity in dynamical systems is important in complex systems research in general, and biological systems modeling in particular. Our method differs from recent community-detection methods that find modularity in static graphs, in that it focuses on the organization of interactions between dynamic system components. In future work, we hope to apply it to the analysis of regulatory and signaling control in biochemical networks, as well as inference of functional neural organization from brain recordings.

### Acknowledgments

Thanks to Randy Beer, Paul Williams, Olaf Sporns, and the participants of the *Guided Self-Organization 3* workshop for useful feedback and encouragement.

### References

- Ay, N. and Polani, D. (2008). Information flows in causal networks. *Advances in Complex Systems*, 11(1).
- Ay, N. and Wennekers, T. (2003). Dynamical properties of strongly interacting Markov chains. *Neural Networks*, 16(10).
- Balduzzi, D. and Tononi, G. (2008). Integrated information in discrete dynamical systems: motivation and theoretical framework. *PLoS Comput Biol*, 4(6).
- Barrett, A., Seth, A., and Sporns, O. (2011). Practical Measures of Integrated Information for Time-Series Data. *PLoS Comput Biol*, 7(1).
- Barron, A. and Hengartner, N. (1998). Information theory and superefficiency. *Ann Stat*, 26(5).
- Bullmore, E. and Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3).
- Conant, R. (1972). Detecting subsystems of a complex system. *IEEE Trans on Systems, Man, and Cybernetics*.
- Dawid, A. (1992). Prequential analysis, stochastic complexity and Bayesian inference. *Bayesian statistics*, 4.
- Force, A., Cresko, W., Pickett, F., Proulx, S., Amemiya, C., and Lynch, M. (2005). The origin of subfunctions and modular gene regulation. *Genetics*, 170(1).
- Guimera, R., Sales-Pardo, M., and Amaral, L. (2004). Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2).
- Han, J. et al. (2004). Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*.
- Hartwell, L. et al. (1999). From molecular to modular cell biology. *Nature*, 402(6761).
- Haussler, D. and Oppen, M. (1997). Mutual information, metric entropy and cumulative relative entropy risk. *Ann Stat*, 25(6).
- Hyvärinen, A. and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5).
- Kashtan, N. and Alon, U. (2005). Spontaneous evolution of modularity and network motifs. *PNAS*, 102(39).
- Komaki, F. (1996). On asymptotic properties of predictive distributions. *Biometrika*, 83(2).
- Krause, A. et al. (2003). Compartments revealed in food-web structure. *Nature*, 426(6964).
- Pearl, J. (2000). *Causality: models, reasoning, and inference*. Cambridge Univ Pr.
- Rissanen, J. (2007). *Information and complexity in statistical modeling*. Springer Verlag.
- Schlosser, G. and Wagner, G. (2004). *Modularity in development and evolution*. University of Chicago Press.
- Simon, H. (1962). The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6).
- Solé, R. and Valverde, S. (2008). Spontaneous emergence of modularity in cellular networks. *J R Soc Interface*, 5(18).
- Strelhoff, C., Crutchfield, J., and Hübner, A. (2007). Inferring Markov chains. *Physical Review E*, 76(1).
- Studený, M. and Vejnarova, J. (1998). The multiinformation function as a tool for measuring stochastic dependence. *Learning in graphical models*, 261.
- Tishby, N., Pereira, F., and Bialek, W. (1999). The information bottleneck method. In *37th Allerton Conf on Communication*.
- Tononi, G., Edelman, G., and Sporns, O. (1998). Complexity and coherency: integrating information in the brain. *Trends in cognitive sciences*, 2(12).
- Tononi, G. and Sporns, O. (2003). Measuring information integration. *BMC Neuroscience*, 4(1).
- Tononi, G., Sporns, O., and Edelman, G. (1994). A measure for brain complexity: relating functional segregation and integration in the nervous system. *PNAS*, 91(11).
- van Erven, T., Grünwald, P., and de Rooij, S. (2007). Catching up faster in Bayesian model selection and model averaging. *NIPS*, 20.
- Wagner, G., Pavlicev, M., and Cheverud, J. (2007). The road to modularity. *Nature Reviews Genetics*, 8(12).
- Winther, R. (2001). Varieties of modules: kinds, levels, origins, and behaviors. *J of Experimental Zoology*, 291(2).



# Reaction Flow Artificial Chemistries

Peter Kreyssig<sup>1</sup> and Peter Dittrich<sup>1</sup>

<sup>1</sup>Bio Systems Analysis Group, Department of Mathematics and Computer Science,  
Friedrich Schiller University Jena, 07743 Jena, Germany  
{peter.kreyssig—peter.dittrich}@uni-jena.de

## Abstract

Artificial chemistries have been analysed mostly under the precondition of a well-stirred reaction vessel. In other words, the localisation of molecules is ignored for simplicity. Here we drop this assumption and replace it with a spatial distribution of molecules given by a flow, i.e. molecules move according to a given vector field. This can be seen as a particular type of dynamics. It also gives additional parameters to the control over the development of the chemistry over time. In particular, the modelling of membranes and transport processes which occur in cells, for example, can be described using continuous vector fields instead of giving a discrete formulation. We give some examples and ideas for analysing such chemistries via a stochastic simulation, a PDE and chemical organisations.<sup>1</sup>

## Introduction

So far many *artificial chemistries* assume a well-stirred reaction vessel (Speroni di Fenizio, 2002). This is a special type of dynamics for the application of rules of the chemistry. In particular, it means that any molecule in the reaction soup can potentially react with any other molecule in the soup at any time. Or in other words, there is no *localisation of molecules* taken into account. On the one hand, this is easier to handle from a technical point of view (Dittrich et al., 2001). In a well-stirred reactor the change of concentration of molecules in the vessel is often seen as a stochastic process and can be simulated using the Gillespie algorithm (Gillespie, 1977, 1976) or can be approximated using ordinary differential equations (ODE), e.g. with the assumption of mass action kinetics. On the other hand, thinking of molecules as not being localised is unrealistic in many situations, e.g. in living cells with their compartments, membranes and transport processes, or when modelling the origin of life (Fishkis, 2010).

There are several approaches to include the spatial organisation or localisation into artificial chemistries. Some of

them employ means of *discrete spatial structures*, like P systems (Păun, 2000), vessels with dynamic compartments using Gillespie's algorithm (Versari and Busi, 2007) or MGS (Giavitto and Michel, 2001), some work with *continuous* additions to the dynamics, like reaction diffusion systems (Adamatzky, 2005). In the first case compartments and their creation and dissolution operations structure the reaction vessel. This gives a discrete description of geometrical information, where the definition of the chemistry is separate from the membrane structure, i.e. the membranes are not formed by molecules. Each of the compartments is subject to a well-stirred stochastic or deterministic dynamics. In the later case, molecules are localised in a Euclidean space. The assumption of a well-stirred domain is replaced by a diffusion process which results in a PDE model. This model describes the dynamics of the artificial chemistry accounting for the reactions and the movement by diffusion. This means that there is no further control over the behaviour possible except of the choice of diffusion constants.

The two examples show that including space into the dynamics brings more *complexity* with it. Still, there is extensive theory at hand for both of them. Going a step further in terms of complexity, we loose this advantage of rigid theoretical descriptions. For example, we have molecular dynamics simulations which can also be combined with rule-based spatial models (Grünert et al., 2010). They use the full generality of possible movement and reactions of molecules at the price of computational costs, predictability and controllability. Furthermore there are approaches, like the swarm chemistry (Sayama, 2009) using space for the representation of molecules.

Here we propose using *vector fields* for modelling spatial organisation and transport processes in artificial chemistries. By this we mean that molecules move along the flow lines of the vector field of a region in  $\mathbb{R}^N$  (most of the time  $N$  will equal to 2 or 3). Reactions are only applicable if enough molecules of the left hand side can be found together close enough. In other words, we do not intend to stir the reactor with our molecules well, but just stir in a particular way defined by a vector field. We can still formulate this as a

<sup>1</sup>further information and videos available at <http://www.biosys.uni-jena.de/Research/Projects/Reaction+Flow+Artificial+Chemistries.html>

stochastic process and approximate it with a partial differential equation (PDE). Another advantage is the gain of control over the behaviour of the chemistry by using different types of vector fields in contrast to, for example, diffusion, where we only have the diffusion coefficients as parameters.

The description of *transport processes* and *membranes* with vector fields, i.e. continuous objects, rather than discrete structures, does not seem to be as convenient or powerful at a first glance. To our understanding membranes and transport are integral part of the dynamics and should therefore be handled with continuous objects fitting in with the usual modelling via ODEs, for example. Therefore we would like to give a proof of concept for non-discrete membranes, compartments and transport. Also we focus here on the spatial aspect rather than on the artificial chemistries and their reactions used.

The paper is organised as follows. First we give the definition of a general reaction flow artificial chemistry. Then a differential model, using a PDE and Mathematica for a numerical computation of solutions is presented. We also describe a stochastic simulator providing us with a tool to run example chemistries. Then we show how to analyse the behaviour of reaction flow artificial chemistries with the help of chemical organisations (Dittrich and Speroni di Fenizio, 2007; Speroni di Fenizio and Dittrich, 2007). Finally we give more examples.

## Reaction Flow Artificial Chemistries

Let  $M$  be a set and  $R$  be a subset of  $\mathcal{P}_{mult}(M) \times \mathcal{P}_{mult}(M)$  where  $\mathcal{P}_{mult}(M)$  denotes the set of multisets over  $M$ . The pair  $(M, R)$  is called *reaction network* and we call  $M$  the set of *molecules* and  $R$  the set of *reactions*.

By *applying* a reaction  $(l, r) \in R$  to a multiset over  $M$  we mean replacing the subset  $l$  by the subset  $r$ . To be able to do so, we assume that the multiset considered is large enough, i.e. that it consists of enough molecules as required on the left hand of the rule.

For  $(l, r) \in R$  we also write  $l \longrightarrow r$  or

$$\sum_{m \in M} l_m m \longrightarrow \sum_{m \in M} r_m m$$

where we denote by  $l_m, r_m \in \mathbb{N}_0$  the multiplicity of  $m$  in  $l, r$  respectively. This resembles notation from chemistry. Furthermore the *support* and the *product* of  $(l, r)$  are

$$\text{supp}(l, r) := \{m \in M \mid l_m > 0\},$$

$$\text{prod}(l, r) := \{m \in M \mid r_m > 0\}.$$

Let  $A$  be a subset of  $M$ . We define  $R_A$ , the set of reaction applicable to  $A$ , by setting

$$R_A := \{(l, r) \in R \mid \text{supp}(l, r) \subseteq A\}.$$

Abusing notation we use a reaction  $(l, r) \in R$  as an index as well and define the *stoichiometric matrix*  $S_A \in \mathbb{R}^{|A| \times |R_A|}$  for  $A$  by

$$(S_A)_{a, (l, r)} = r_a - l_a, \quad a \in A, (l, r) \in R.$$

If we add to a reaction network  $(M, R)$  a domain for the molecules and an algorithm that determines how the rules are applied to the molecules within the domain, we get an *artificial chemistry* (Dittrich et al., 2001). For a *reaction flow artificial chemistry* we choose a region  $U$  in  $\mathbb{R}^N$  as the domain. The elements of an initial multiset are placed in this region. Molecules can then only react if they are “suitably” close. What this means exactly is of course to be defined from case to case. Here, we will always choose a small number for the maximum distance in which molecules can still react. Additionally, molecules change their position according to the flow lines of a given *vector field*  $V : U \rightarrow \mathbb{R}^N$ . This means that in one iteration of the algorithm a molecule at position  $\mathbf{x} \in \mathbb{R}^N$  changes to the position  $\mathbf{x} + V(\mathbf{x})$ . We assume that the new position is again in  $U$ . The described way of movement can be interpreted as mixing molecules according to a fixed scheme or algorithm. In contrast to a reaction diffusion system even after a long time period there is no guarantee that the multiset of molecules will be stirred well.

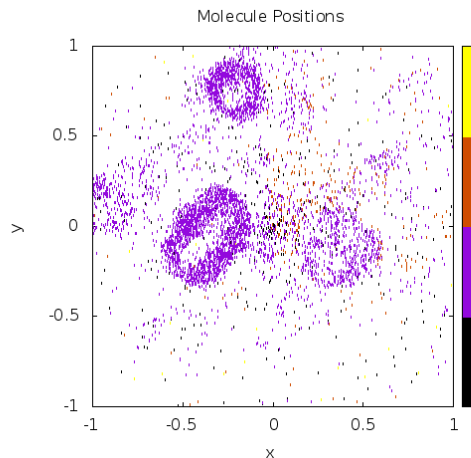


Figure 1: Stochastic simulation of the reaction flow artificial chemistry  $(M_1, R_1)$  in the region  $U_1$  with vector field  $V_1$  after 400 iterations, for details see Section II. Species are marked with colours from black for  $a$  to yellow for  $d$ .

Figure 1 shows the state of the following reaction flow artificial chemistry after 400 iterations. The reaction network we chose is  $M_1 = \{a, b, c, d\}$  with

$$R_1 = \{a + b \rightarrow a + 2b, a + d \rightarrow a + 2d, b + c \rightarrow 2c, c \rightarrow b, b + d \rightarrow c, b \rightarrow \emptyset, c \rightarrow \emptyset, d \rightarrow \emptyset\}.$$

As a domain  $U_1 = [-1, 1] \times [-1, 1]$  is used and a starting set of molecules of size 2500 is placed randomly around  $(0, 0)$ . The vector field responsible for the movement is a swirl given by

$$V_1(x, y) = \frac{1}{\sqrt{2}} \left( (1 - \sqrt{2})x - y, x + (1 - \sqrt{2})y \right),$$

see Figure 2.

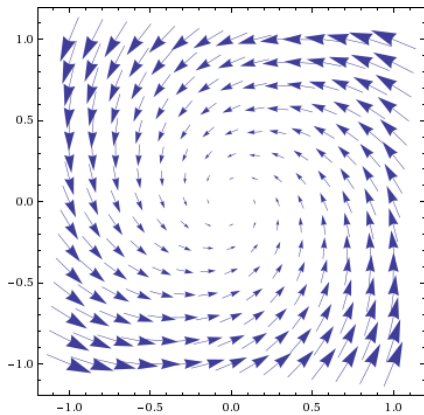


Figure 2: The vector field  $V_1$  in the region  $U_1 = [-1, 1] \times [-1, 1]$ . The length of vectors is scaled by 0.3 for a better readability.

The parameters of the simulation (for the detailed description see Section IV) we set to  $n = 1600$ ,  $s = 400$ ,  $rad = 1.0$ ,  $u = 1.0$  and  $r = 0.1$ .

We see a development of patches of the species  $b$  over time. This is a rather different behaviour compared to the same reaction network, when investigated using a well-stirred system or diffusion. When we do a simulation using these dynamics in our particular case of the reaction network  $(M_1, R_1)$ , the system is completely described by concentration alone without taking position of molecules into account.

We give some ideas for slight *generalisations* of this approach, though they are not used here. Separate flows for different molecular species can be used, i.e. there is a set of fields  $V_{m_1}, \dots, V_{m_{|M|}}$  such that each field is responsible for the movement of a single species. Maybe this makes sense if, for example, taking the different weight of molecular species or semi-permeable membranes into account. Also the vector field(s) could be time depended, i.e. there is a dynamical change of the transport of molecules over time. Another interesting way of extending the concepts is by letting the underlying reaction network influence the vector field, e.g. particular types of (bigger) molecules could block (smaller) other ones.

We investigate two models for this kind of artificial chemistry. The first one is a PDE describing the continuous change of concentration of the molecules. The second one is a stochastic simulation of the movement of molecules in the flow and the reactions they take part in.

## Differential Model

We concentrate on the case  $N = 2$  to keep it simple, but still easily generalisable. For the differential model we assume that every point  $(x, y) \in \mathbb{R}^2$  bares concentrations of all the species  $M = \{m_1, \dots, m_{|M|}\}$ . The concentration of a species  $m_i$ ,  $i \in \{1, \dots, |M|\}$  in  $(x, y)$  at time  $t \geq 0$  is  $[m_i](x, y, t)$ , so

$$[m_i] : \mathbb{R} \times \mathbb{R} \times \mathbb{R}^+ \rightarrow \mathbb{R}.$$

For readability we omit the coordinates and write simply  $[m_i]$ .

We describe the change of concentration over time with two summands. The change of molecule concentration given by the vector field  $V$  is the *directional derivative* of  $[m_i] \cdot \|V\|$  in the direction of  $V$ . This is exactly the formalisation of the statement that molecules follow the flow lines of the vector field. The change caused by the reactions is summarised in the *reaction terms*  $R_{\mathbf{k},i}$ . They depend on the concentrations of all molecules  $[m_1], \dots, [m_{|M|}]$  and the constant reaction rates  $\mathbf{k} \in \mathbb{R}^{|R|}$ .

When assuming the mass action kinetics for the dynamics of the reaction network  $(M, R)$ , we can write down the reaction terms as follows. Let us denote the *flux vector function* by

$$v_{M,\mathbf{k}} : \mathbb{R}_{\geq 0}^{|M|} \rightarrow \mathbb{R}_{\geq 0}^{|R|}.$$

Still abusing notation we use a reaction  $(l, r) \in R$  as an index as well and define

$$(v_{M,\mathbf{k}}([m_1], \dots, [m_{|M|}])(l, r)) = \mathbf{k}_{(l,r)} \prod_{i=1}^{|M|} [m_i]^{l_{m_i}}.$$

The  $i$ th reaction term is the  $i$ th component of the vector yielded by the product of the stoichiometric matrix with the flux vector function,

$$R_{\mathbf{k},i}([m_1], \dots, [m_{|M|}]) = (S_M \cdot v_{M,\mathbf{k}}([m_1], \dots, [m_{|M|}])(l, r))_i.$$

The equation defining the behaviour of the reaction flow artificial chemistry is

$$\frac{\partial [m_i]}{\partial t} = -\frac{1}{\|V\|} \langle \nabla ([m_i] \|V\|), V \rangle + R_{\mathbf{k},i}([m_1], \dots, [m_{|M|}])$$

where the gradient  $\nabla = \nabla_{(x,y)}$  is taken for  $(x, y)$ ,  $\|V\| = \|V(x, y)\|_2$  is the Euclidean norm of the field and  $\langle \cdot, \cdot \rangle$  denotes the Euclidean scalar product.

In the case of an integrable field, i.e. there is  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  with  $\nabla f = V$ , the molecules follow a gradient flow to the sinks of the function.

As a simple example we numerically solve the equation for the reaction flow artificial chemistry given by the reaction network  $(M_1, R_1)$  as defined before with the radially symmetric field

$$V_2(x, y) = \frac{\cos 10r}{10r}(x, y),$$

i.e. the above case with  $f(x, y) = -0.01 \sin 10r$ , where we abbreviate  $r = \|(x, y)\|_2 = \sqrt{x^2 + y^2}$ . We assume mass action kinetics with all reaction constants equal to 1.

Since the directional derivative is independent of the chosen coordinate system and since our field is radially symmetric, it suffices to solve the equation for one spatial dimension. Therefore we solve the equation dependent on  $r$ . The solution on  $\mathbb{R}^2$  is then the solution we get in the one dimensional case extended to  $\mathbb{R}^2$ , i.e. we apply it to the distance  $\sqrt{x^2 + y^2}$ . We arrive at the equations

$$\begin{aligned}\frac{\partial[a]}{\partial t} &= -0.1 \frac{\partial[a] \cos 10r}{\partial r} \\ \frac{\partial[b]}{\partial t} &= -0.1 \frac{\partial[b] \cos 10r}{\partial r} + [a][b] - [b][c] + [c] - [b][d] - [b] \\ \frac{\partial[c]}{\partial t} &= -0.1 \frac{\partial[c] \cos 10r}{\partial r} + [b][c] - [c] + [b][d] - [c] \\ \frac{\partial[d]}{\partial t} &= -0.1 \frac{\partial[d] \cos 10r}{\partial r} + [a][d] - [b][d] - [d].\end{aligned}$$

This can be numerically solved. We assume an initial constant concentration of 1 for all species and use Mathematica's NDSolve to get Figure 3 compared to the stochastic simulation Figure 4.

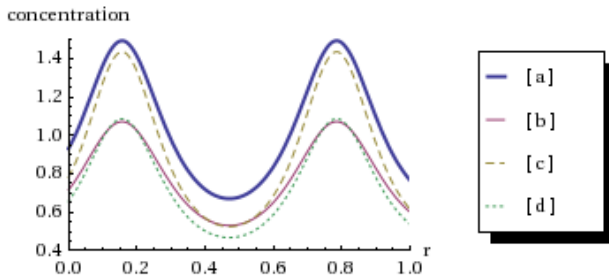


Figure 3: Numerical solution to the equation system at time 0.4 in one dimension.

## Stochastic Model

Additionally to the rather theoretical approach via a PDE we also implemented a stochastic simulator for the reaction flow artificial chemistries. This allows us to run some concrete examples.

We assume a reaction network  $(M, R)$  is given. As the domain or region the set  $[-1, 1] \times [-1, 1] \subset \mathbb{R}^2$  is used for all our examples even though the size of the square is variable.

Different ways of initially placing molecules can be used. In the examples shown here, we initially place  $n$  molecules randomly around the origin in a circle of radius  $rad$ . More precisely, in Figure 4 we choose random coordinates  $x$  and  $y$  such that  $\sqrt{x^2 + y^2} < rad$  to achieve a uniform distribution of molecules. In the other examples we choose a random

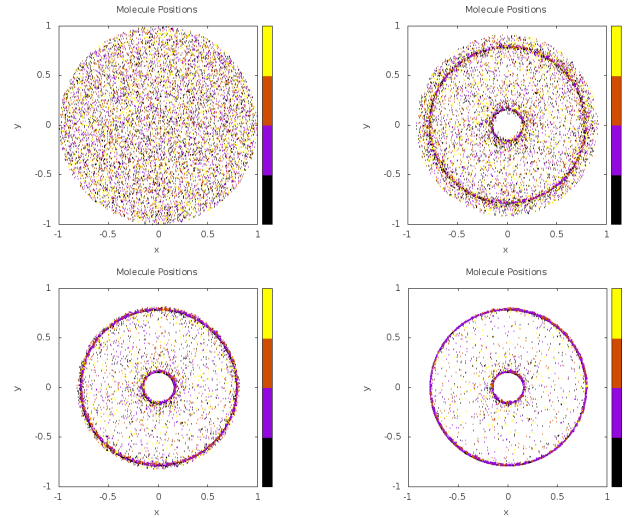


Figure 4: Stochastic simulation of the reaction flow artificial chemistry  $(M_1, R_1)$  in the region  $U_1$  with vector field  $V_2$  after 0, 1, 2 and 3 iterations. Parameters are  $n = 10000$ ,  $rad = 1.0$ ,  $u = 1.0$  and  $r = 0.01$ .

angle between 0 and  $2\pi$  and a random length between 0 and  $rad$  to position them. Each starting molecule is assigned a random type of species.

For  $s$  simulation steps we apply the vector field  $V$  and the rules of  $R$  in the following manner. A molecule at position  $(x, y)$  is moved to position  $(x, y) + V(x, y)$ . As mentioned before, we assume that the new molecule position is again in our region  $[-1, 1] \times [-1, 1]$ . If this is not the case, we can use cyclic or solid boundary conditions or increase the size of the region. Typically several vector fields are added up or are applied at suitable parts of the domain region to account for different effects in time and space to generate the required behaviour.

Then the reaction rules are applied to a randomly chosen  $u$  percent of the molecules present in the domain. For each chosen molecule  $m$  with, we assume, position  $(x, y)$  we look at neighbouring molecules, i.e. molecules with no more than distance  $r$  to  $m$ . Let  $A_r(m)$  be this multiset of molecules found in

$$U_r(m) = \{(x', y') \mid \text{dist}((x', y'), (x, y)) < r\} \subset \mathbb{R}^2.$$

There are several different ways of applying rules to  $A_r(m)$  possible. For the examples given here only the following is used. A number of  $|R|$  reaction rules are randomly chosen from  $R$ . It is checked whether they are applicable and if so applied to the multiset  $A_r(m)$ . By this we mean that if molecules have to be removed, they vanish from the domain and if they are added they are positioned randomly in  $U_r(m)$ .

## Analysis through Chemical Organisations

A subset  $A$  of  $M$  is *closed* if for all reactions  $(l, r) \in R_A$  we have  $\text{prod}(l, r) \subseteq A$ , i.e. if  $(A, R_A)$  is a reaction network.  $A$  being closed means that by applying reactions from  $R_A$  to multisets over  $A$  we do not get molecules outside  $A$ .

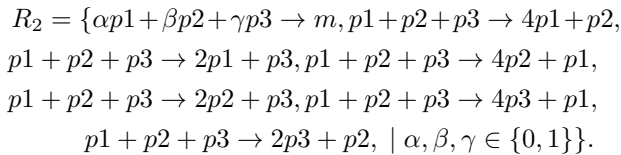
A subset  $A$  of  $M$  is *self-maintaining* if there is a vector  $v \in \mathbb{R}^{|R_A|}$  with strictly positive entries such that  $S_A v \in \mathbb{R}^{|A|}$  has only non-negative entries.  $A$  being self-maintaining means that applying reactions from  $R_A$  at certain rates to a multiset over  $M$  does not reduce the number of molecules of any species of  $A$ .

A subset of  $M$  is a *chemical organisation* (Dittrich and Speroni di Fenizio, 2007) if it is closed and self-maintaining. The set of organisations is called  $\mathcal{O}$ .

As proposed in (Speroni di Fenizio and Dittrich, 2007) we can look at the chemical organisations at different spatial scales at different times. The idea is to identify functional units when looking at the development of a chemistry in the domain over time. Only the organisations, as the closed and self-maintaining sets, are able to stay in the domain for a longer time period. Therefore persistent structures should be an organisation. This can also be interpreted as identifying higher level units.

In the described stochastic model we looked for organisations in the following way. The domain is divided into squares of size  $orgRad$ . The species present in each square are collected and then the biggest organisation contained in this set is computed. In the examples presented here  $orgRad$  is 0.1.

As an example for this analysis via organisations we demonstrate the formation of a membrane, see Figure 5. The used reaction flow artificial chemistry is defined by  $M_2 = \{m, p1, p2, p3\}$ ,



This reaction network is constructed such that arbitrary combinations of the producer molecules  $p1, p2, p3$  build the membrane molecule  $m$ . The other reactions account for the rebuilding of the producers over time if enough of all three different species  $p1, p2, p3$  are present. The desired behaviour corresponds to the organisations  $\mathcal{O} = \{\emptyset, \{m\}, \{m, p1, p2, p3\}\}$  the reaction network  $(M_2, R_2)$  exhibits. In this example we can think of  $\{m\}$  as the representation of the membrane and of  $\{m, p1, p2, p3\}$  as the membrane producing core. The vector field is defined by

$$V_2(x, y) = \begin{cases} 0.0005 \frac{e^{5r}}{r}(x, y) & 0.2 < r < 0.8 \\ V_1(x, y) & \text{else} \end{cases}$$

where  $r = \sqrt{x^2 + y^2}$  and  $V_1$  is the earlier defined field. The field accounts for a mixing close to the origin, a transport away from it and a movement of the membrane. All molecular species are transported by the field, but the rules are constructed such that most of the producer molecules  $p1, p2, p3$  are destroyed on their way to the membrane. Of course, we cannot guarantee that none of them appears in the membrane built by molecules of type  $m$ . The analysis via chemical organisations suggests that even if they make it to the membrane, they will not be able to stay for long, see Figure 6. When using separate flows for different molecular species, the formation of a membrane is even easier realised.

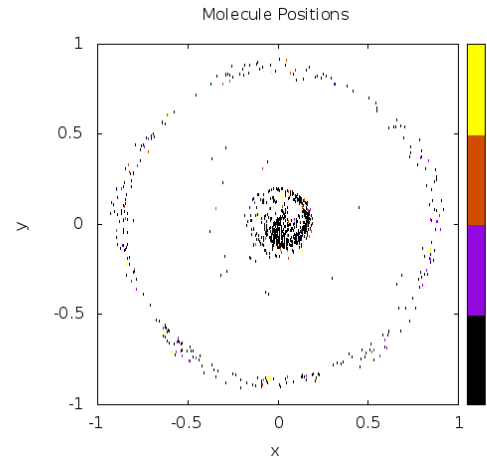


Figure 5: A core emitting molecules which form a membrane around the core. State after 100 iterations. Parameters are  $n = 2500$ ,  $rad = 0.2$ ,  $u = 1.0$  and  $r = 0.15$ .

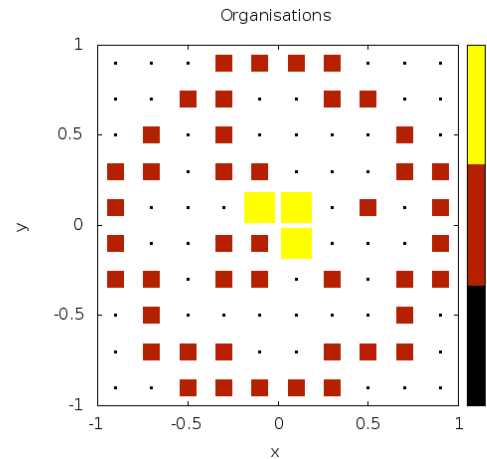


Figure 6: Analysis via chemical organisations. The biggest organisation  $\{m, p1, p2, p3\}$  (yellow) shows primarily in the core, the smaller one  $\{m\}$  (red) as the membrane. State after 100 iterations.



## Further Experiments and Examples

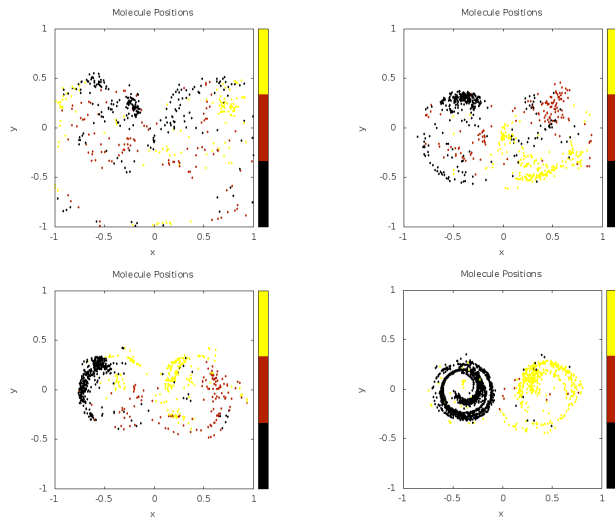


Figure 7: Formation of two compartments. State after 5, 15, 30 and 150 iterations. Parameters are  $n = 2500$ ,  $rad = 0.8$ ,  $u = 1.0$  and  $r = 0.1$ .

**Formation of Compartments.** This example shows the slow formation of two compartments in a domain for a reaction network of three competing species. Initially the molecules are distributed over the region. Due to a vector field pushing them to the left and right hand side respectively they gather in two different areas where they are stirred by another two fields, see Figure 7. The parameters for the simulation are as follows. The network is taken from (Neumann and Schuster, 2007) as a model for the rock-scissor-paper game. There are three different competing species present  $M_3 = \{s_1, s_2, s_3\}$  with the reactions

$$R_3 = \{s_i \rightarrow 2s_i, 2s_i \rightarrow s_i, \\ s_i + s_j \rightarrow s_i, s_i + s_j \rightarrow s_j \mid i \neq j\}.$$

The vector field, see Figure 8, is given by

$$V_3(x, y) = \begin{cases} -0.01e^{5x}(x, 0) + V_1(x + 0.4, y) & x < 0 \\ 0.01e^{-5x}(x, 0) + V_1(x - 0.4, y) & x > 0 \end{cases}$$

Similar to  $V_2$  in the last section the first part of  $V_3$  accounts for the pushing of molecules away from the centre. The second part is a shifted swirl, as described in Section II.

**Emergent Behaviour.** In this example we use the same vector field  $V_3$  as before, but with a different reaction network ( $M_4, R_4$ ) and parameters. The network is the central sugar metabolism of *Escherichia coli* as described in (Puchalka and Kierzek, 2004) with the adaptations made in (Centler et al., 2007). We do not give the full definition here due to the size of the chemistry,  $|M_4| = 92$  and  $|R_4| = 198$ .

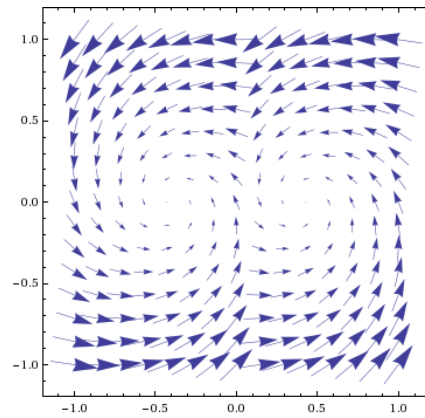


Figure 8: The vector field  $V_3$  in the region  $[-1, 1] \times [-1, 1]$ . The length of vectors is scaled by 0.3 for a better readability.

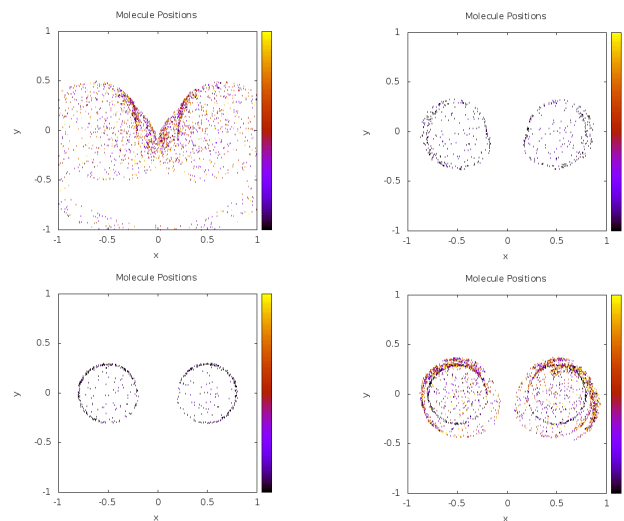


Figure 9: Emergent behaviour. State after 5, 15, 65 and 70 iterations. Parameters are  $n = 2500$ ,  $rad = 1.0$ ,  $u = 1.0$  and  $r = 0.15$ .

Here due to the movement of molecules an unexpected effect happens, in particular unexpected from the chemical organisation point of view. From Figure 9 we see that after 15 iterations (second picture) the chemistry seems to stabilise since till iteration 65 (third picture) no qualitative change happens. There are formed two rings of many molecules of one species. Then, due to the transport, there are reactions possible again, so that a qualitative change seems to happen (fourth picture). Molecules of other species are build again swirl around the ring and vanish after some more time.

## Conclusion and Outlook

We suggested a new approach for introducing a structured space into the dynamics of an artificial chemistry. This is done by using a vector field to generate a flow of molecules

located in a Euclidean space. The introduction of vector fields as defining part for the movement of molecules in space gives additional parameters for the control of the dynamics. We have shown that by this means we can describe membranes, membrane channels and transport processes.

Our system can now be applied to study the influence of different flow structures on the *evolvability* of a chemical system. It is known that compartmentalisation is in a certain sense beneficial for pre-biotic (chemical) evolution (Fernando and Rowe, 2007). Since in a pre-biotic scenario various flow structures were likely present (Martin et al., 2008), it would be interesting to study whether and how particular flow structures can lead to “improved” chemical evolution. Note that space has already a positive effect when just assuming diffusion by counteracting on parasitism (Boerlijst and Hogeweg, 1991; Fishkis, 2010). But could a flow structure add further evolutionary benefits?

Another direction of research could investigate the role of different flow structures for *bio-chemical information processing*. Does a particular flow contribute additional information processing capability to those of reaction-diffusion systems (Adamatzky, 2005)? For a given (artificial) chemistry, we could evolve the flow instead of the chemistry itself. By doing so, we could study the role of flow for certain functions, separated from the reactions going on. This could have practical implications in the development of novel bio-chemical information technologies, since, it should be easier to change the flow, e.g. within a microfluidic system, than the chemistry.

Finally we can use the scenario presented here to extend the notion of a *spatial chemical organization* (Speroni di Fenizio and Dittrich, 2007) including flow and diffusion.

## Acknowledgement

We thank Gabi Escuela, Stephan Peter and Gerd Grünert very much for helpful discussions and support. We also acknowledge financial support by the German Research Foundation (DFG) Grant DI 852/4-3.

## References

- Adamatzky, A. (2005). Programming reaction-diffusion processes. *Unconventional Programming Paradigms*, pages 33–46.
- Boerlijst, M. C. and Hogeweg, P. (1991). Spiral wave structures in prebiotic evolution: Hypercycle stable against parasites. *Physica D*, 48:17–28.
- Centler, F., Fenizio, P., Matsumaru, N., and Dittrich, P. (2007). Chemical organizations in the central sugar metabolism of *Escherichia coli*. *Mathematical Modeling of Biological Systems, Volume I*, pages 105–119.
- Dittrich, P. and Speroni di Fenizio, P. (2007). Chemical organization theory. *Bulletin of Mathematical Biology*, 69(4):1199–1231.
- Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial chemistries—a review. *Artificial Life*, 7(3):225–275.
- Fernando, C. and Rowe, J. (2007). Natural selection in chemical evolution. *Journal of Theoretical Biology of the Cell*, 247(1):152–167.
- Fishkis, M. (2010). Emergence of Self-Reproduction in Cooperative Chemical Evolution of Prebiological Molecules. *Origins of Life and Evolution of Biospheres*, pages 1–15.
- Giavitto, J. and Michel, O. (2001). MGS:: A Rule-Based Programming Language for Complex Objects and Collections. *Electronic Notes in Theoretical Computer Science*, 59(4):286–304.
- Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J Comp Phys*, 22(4):403–434.
- Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *J Phys Chem*, 81(25):2340–2361.
- Grünert, G., Ibrahim, B., Lenser, T., Hinze, T., and Dittrich, P. (2010). Rule-based spatial modeling with diffusing, geometrically constrained molecules. *BMC Bioinformatics*, 11:307.
- Martin, W., Baross, J., Kelley, D., and Russell, M. J. (2008). Hydrothermal vents and the origin of life. *Nat. Rev. Microbiol.*, 6:805–814.
- Neumann, G. and Schuster, S. (2007). Continuous model for the rock–scissors–paper game between bacteriocin producing bacteria. *Journal of Mathematical Biology*, 54(6):815–846.
- Păun, G. (2000). Computing with membranes. *J. Comput. Syst. Sci.*, 61(1):108–143.
- Puchalka, J. and Kierzek, A. (2004). Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks. *Biophysical Journal*, 86(3):1357–1372.
- Sayama, H. (2009). Swarm chemistry. *Artificial Life*, 15(1):105–114.
- Speroni di Fenizio, P. (2002). Artificial chemistries. *Bulletin of the European Association for Theoretical Computer Science*, 76:128–141.
- Speroni di Fenizio, P. and Dittrich, P. (2007). Chemical organizations at different spatial scales. In e Costa, F. A., Rocha, L. M., Costa, E., Harvey, I., and Coutinho, A., editors, *Proc. European Conference on Artificial Life (ECAL07)*, volume 4648 of *LNCIS*, pages 1–11. Springer, Berlin.
- Versari, C. and Busi, N. (2007). Stochastic simulation of biological systems with dynamical compartment structure. In *Computational Methods in Systems Biology*, pages 80–95. Springer.



# Artificial Organelle for Energy Production in Artificial Cell

Yutetsu Kuruma<sup>1</sup>, Toshiharu Suzuki<sup>2</sup>, Masasuke Yoshida<sup>3</sup>, Takuya Ueda<sup>1</sup>

<sup>1</sup>Department of Medical Genome Sciences, Graduate School of Frontier Sciences, The University of Tokyo, Japan

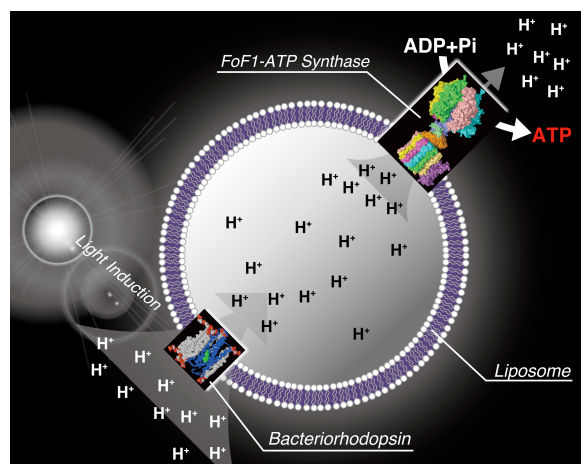
<sup>2</sup>ATP Synthesis Regulation Project, ICORP, Japan Science and Technology Corporation, Japan

<sup>3</sup>Department of Molecular Biosciences, Kyoto Sangyo University, Japan

kuruma@molbio.t.u-tokyo.ac.jp

## Abstract

Autonomous production of biological energy is one of the primal processes in living cells. In cells, the bio-energy presents as adenosine or guanosine triphosphate (ATP or GTP) is used for the most of cellular reactions. ATP is produced through the glycolytic cycle by a series of dedicated enzymes in cytosol, or through the oxidative phosphorylation of adenosine diphosphate (ADP), operated by ATP synthase, which is located on lipid membrane. For instance, in mitochondria, the proton potential across the membrane, generated by an electron transport chain, is eventually dissolved through FoF1-ATP synthase (FoF1). The flux of protons drives FoF1 and activates the synthesis of ATP, from ADP and Pi. On the other hand, bacteriorhodopsin (bR) is widely known as proton pump machinery that transports the protons to the other side of membrane due to light stimulation. Therefore, our idea is that if the bR and FoF1 were synthesized on a liposome membrane, the resulting liposome is able to generate ATP (see Figure 1).



**Figure 1.** Schematic of bR-FoF1 liposome. Light induced bacteriorhodopsin pumps  $H^+$  into liposome and produced  $H^+$  gradient is used for FoF1-ATP synthase to synthesize ATP.

Racker and Stoeckenius (1) have studied a model system by combining purple membrane, which contains bR, and isolated ATP synthase in phospholipid vesicles. In order to design and construct a synthetic cell in the synthetic biology context, this kind of “bioreactor” should be autonomously

built up through an internal metabolic process such as gene expression. If both bR and FoF1 proteins were synthesized in the presence of organelle-sized vesicles, it would be possible to construct *in vitro* the bR-FoF1 liposomes as a consequence of the artificial protein synthesis and the self-organization of the synthesized proteins. Additionally, so produced bR-FoF1 liposomes can be applied as a bioenergy-producible plant that activates further biological reaction. For instance, if the produced ATP could be used for protein synthesis reaction, the whole system would represent an energetically independent autonomous system.

In the ECAL11 meeting, we present some experimental achievements toward the construction of bR-FoF1 liposome (2, 3). Our recent results show that bR was synthesized in a cell-free protein synthesis system (4) in the presence of liposomes and all-*trans* retinal. Fo complex, the membrane integrated part of FoF1, was synthesized *in situ* and formed the desired FoF1 complex in combination with a supplied F1. FoF1 complex was fully functional, by showing ATPase driven  $H^+$ -translocation activity. These results imply that the bottom up construction of an artificial organelle, which is capable of generating the bioenergy, is experimentally feasible. We believe that our bR-FoF1 liposomes will be essential machinery for constructing artificial cells.

## References

1. Racker, E. and Stoeckenius, W. (1974). Reconstitution of purple membrane vesicles catalyzing light-driven proton uptake and adenosine triphosphate formation. *J. Biol. Chem.*, 249 :662-663.
2. Kuruma, Y., Suzuki, T., Ueda, T. (2010). Production of multi-subunit complexes on liposome through an E. coli cell-free expression system. *Methods Mol Biol.* 607:161-171.
3. Ozaki, Y., Suzuki, T., Kuruma, Y., Ueda, T., Yoshida, M. (2008). Uncl protein can mediate ring-assembly of c-subunits of FoF1-ATP synthase *in vitro*. *Biochem Biophys Res Commun.* 367:663-666.
4. Shimizu, Y., Inoue, A., Tomari, Y., Suzuki, T., Yokogawa, T., Nishikawa, K., and Ueda, T. (2001). Cell-free translation reconstituted with purified components. *Nat. Biotechnol.*, 19:751-755.

# Genetic Transposition Inspired Incremental Genetic Programming for Efficient Coevolution of Locomotion and Sensing of Simulated Snake-like Robot

Tüze Kuyucu<sup>1</sup>, Ivan Tanev<sup>1</sup>, and Katsunori Shimohara<sup>1</sup>

<sup>1</sup>Department of Information Systems Design, Doshisha University, Kyotanabe, Japan  
{tkuyucu, itanev}@mail.doshisha.ac.jp

## Abstract

Genetic transposition (GT) is a process of moving sequences of DNA to different positions within the genome of a single cell. It is recognized that the transposons (the jumping genes) facilitate the evolution of increasingly complex forms of life by providing the creative playground for the mutation where the latter could experiment with developing novel genetic structures without the risk of damaging the already existing, well-functioning genome. In this work we investigate the effect of a GT-inspired mechanism on the efficiency of genetic programming (GP) employed for coevolution of locomotion gaits and sensing of the simulated snake like robot (Snakebot). In the proposed approach, the task of coevolving the locomotion and the sensing morphology of Snakebot in a challenging environment is decomposed into two subtasks, implemented as two consecutive evolutionary stages. At first stage we employ GP to evolve a pool of simple, sensorless bots that are able to move fast in a smooth, open terrain. Then, during the second stage, we use these Snakebots to seed the initial population of the bots that are further subjected to coevolution of their locomotion control and sensing in a more challenging environment. For the second phase the seed is used as it is to create only part of a new individual, and the rest of the new individual's genetic makeup is created by a mutant copy of the seed. Experimental results suggest that the proposed two-staged GT inspired incremental evolution contributes to significant increase in the efficiency of the evolution of fast moving and sensing Snakebots.

## Introduction

Snake-like robots feature potential robustness characteristics beyond the capabilities of most wheeled and legged vehicles, such as: the ability to traverse challenging terrain and insignificant performance degradation when partial damage is inflicted. Some useful features of snake-like robots include smaller size of the cross-sectional areas, stability, ability to operate in difficult terrain, good traction, and complete sealing of the internal mechanisms (Dowling, 1999; Hirose, 1993). Moreover, due to the modularity of their design, the snake-like robots feature high redundancy and fault tolerance (Tanev et al. 2005). Robots with such properties can be valuable for applications that involve exploration, reconnaissance, medicine and inspection.

Designing a controller that can achieve optimal locomotion of a modular Snakebot is a challenging task due to the large number of degrees of freedom in the movement of segments of a Snakebot. The locomotion gait of such bots is often seen as an emergent property; observed at a higher level of consideration of complex, nonlinear, hierarchically organized systems, comprising many relatively simply-defined entities

(morphological segments). In such complex systems the higher-level properties of the system and the lower-level properties of comprising entities cannot be directly induced from each other (Morowitz, 2002). Therefore even if an effective incorporation of sensing information in fast and robust locomotion gaits might emerge from intuitively defined sensing morphology and simple motion patterns of morphological segments, neither the degree of optimality of the developed code nor the way of how to incrementally improve this code is evident to the human designer (Koza et al. 2000). The previous research demonstrates that the control for a fast moving modular robotic organism could be automatically developed through various nature-inspired paradigms, based on models of learning and evolution. The earlier work demonstrates the use of GP (Koza, 1994) for evolution of sensorless sidewinding Snakebots in various environmental conditions (Tanev et al. 2005). Furthermore, the coevolution of active sensing and the control of the locomotion gaits of Snakebots was achieved (Tanev and Shimohara, 2008). The morphology of the sensors, attached to each of the segments of the bot, coevolve with the way to incorporate the sensory readings into the control of locomotion of the bot. The genetically optimized morphological traits of the bot include the initial orientation, the timing of switching on, and the range of the simulated laser range finders (LRF) attached to each of the segments of the bot. The emergent features of the evolved gaits include both the contact and contactless wall-following navigation accomplished via adaptive, sensory-controlled differential steering of the fast moving sidewinding bot. Despite the abovementioned evidence of the feasibility of coevolution of active sensing and the locomotion, the resulting wall-following behavior is achieved in an environment that is too simplified, and therefore too distant from the real-world applications: a simple curved corridor with a plain, smooth surface.

In this work we further investigate the coevolution of the active sensing and locomotion control of sidewinding Snakebot in a more complex environment that, in addition to a narrow corridor, features several large obstacles and many randomly placed small obstacles constituting a rugged terrain within this challenging environment. The sensors on the Snakebot used in this paper follow the same model as proposed in (Tanev and Shimohara, 2008): each segment of the Snakebot is provided with a fixed, immobile LRF with evolvable initial orientation, range and timing of firing. Thus the evolutionary task is not only to determine the time patterns of turning angles and the incorporation of sensor values for effective sensing and locomotion, but also to optimize the

initial orientation, effective range and the timing of activation of module sensors. Hence, the Snakebot genotype is represented as a triple consisting of a linear chromosome containing the encoded values of the three relevant parameters of LRF, and two parse trees corresponding to the algebraic expressions of the temporal patterns of the desired turning angles in horizontal and vertical directions (further detailed in Section “Algorithmic Paradigm”). The most efficient locomotion gaits of Snakebot are not necessarily associated with the forward, rectilinear motions (and sidewinding might emerge as a fast and robust locomotion). Therefore, the eventual fusion of the readings of many sensors mounted in all the segments of the bot would provide Snakebot with the capability to perceive the features of surrounding environment along its whole body. In addition to the widening of the area of the perceived surroundings, multiple sensors offer the potential advantages of robustness to damage of some of them, dependability of the sensory information, and an ability to perceive the spatial features of the surrounding environment due to the motion parallax.

The poor scalability is a common problem in the simultaneous evolution of multiple features of simulated creatures, as the search space of evolution increases faster than linearly with the increase of the number of simultaneously evolved features. The considered case of Snakebot implies that the size of evolutionary search space can be seen as a multiplication of the sizes of the search spaces of the following interdependent evolutionary subtasks:

- Evolution of *control of locomotion*: the time patterns of turning angles of actuators that result in a fast locomotion of the bot,
- Evolution of the *morphology of the active sensing* – initial orientation of the sensors, their range, and timing of their activation, and
- Evolution of the *incorporation of the sensor signals* into the control of locomotion of the bot.

The large search space of the evolution of the considered Snakebot results in an intractable computational effort. Therefore, we propose an approach of decomposing the initially defined task into two subtasks, implemented as two consecutive evolutionary stages. As the first stage we employ GP to evolve a pool of simple, generic sensorless bots that are able to move fast in a smooth, plain terrain. During the second stage, we use these Snakebots to seed the initial population of the bots that are further subjected to coevolution of their locomotion control, sensing morphology, and the method of incorporating the sensor signals into the locomotion of the bot in the given environment.

In this paper we propose an incremental evolution through the elaborated two stages, interfaced by a new approach to seeding. Inspired by genetic transposition (GT), we use the seed from the bots evolved during the first stage to create only a part of a new individual in the second stage. The rationale for proposing such an approach is based on the observations that the evolved fast moving Snakebots with sensory abilities exhibit some emergent locomotion traits that are pertinent to the generic, sensorless sidewinding locomotion (Tanev and Shimohara, 2008). We speculate that a better computational efficiency of evolution can be achieved if we first allow these generic features to evolve in sensorless bots moving in a smooth, plain terrain (with the task featuring a narrow

evolutionary search space), and then–incorporating the genotypes of these bots into the evolution of the morphologically more complex bots (with sensors) in a challenging environment. The proposed mechanism of incorporation of these generic features of locomotion is based on seeding the initial population of GP (employed for the evolution of the bot with sensors) via the GT-inspired mechanism. Using the proposed mechanism of GT, the seed does not form the whole genome of an individual Snakebot, but only a part of it. We believe that, similar to the nature, the latter would offer the opportunity to preserve the genetic makeup of the generic locomotion features intact, while incrementally “upgrading” it with the new sensing abilities.

From another perspective, our work is inspired by the discoveries in the neurobiology suggesting that the complex navigation behaviors of species in nature can be achieved through an appropriate real-time modulation, controlled by the sensory inputs, of the generic neural signals produced by sensorless central pattern generators (CPG) (Levitin and Kazczmarek, 2002). Within this context, we would like to investigate whether (i) the separation of the genotype into two parts, mimicking the natural CPG and its modulation via sensory processing, respectively, and (ii) evolving these two parts in two consecutive stages would contribute to the improvement of the efficiency of evolution of the Snakebot.

In the remaining of this document we will provide a brief background related to GT, followed by a section elaborating on both the evolutionary and the experimental frameworks used in this paper. Next, we will discuss the obtained experimental results, and finally draw the conclusions of the work presented and detail future work.

## Genetic Transposition in GP

Discovered by Barbara McClintock in maize (*Zea mays*), the *transposons* (jumping genes) are sequences of DNA that can move around to different positions within the genome of a single cell, in a mechanism called *transposition* (McClintock, 1950). In the process, they can cause mutations and change the amount of DNA in the genome. It is recognized that the transposons, facilitate the evolution of increasingly complex forms of life by providing the creative playground for fast mutations where the latter could experiment with developing novel genetic structures without the risk of damaging the already existing, well-functioning genome (Nowacki et al. 2009; Strand and McDonald, 1985).

The related transposition-inspired research in evolutionary computation (EC) started by the work of Simoes and Costa (Simoes and Costa, 1999; Simoes and Costa, 2000) on the favorable effect of transposition on the performance of genetic algorithms (GA). The first of their methods is intended to enhance the crossover operation in GA by exchanging only the genetic material that is specifically marked as a transposon (Simoes and Costa, 1999). Their second approach, (termed “asexual transposition”) models the mutation of GA as a “cut and paste” operation observed in biological GT (Simoes and Costa, 2000). Chan et al demonstrate a successful implementation of a GT inspired mechanism in multi-objective optimization, which is shown to have superior performance in achieving pareto optimal solution in comparison to multi-objective optimization without the GT

mechanism (Chan et al, 2008). Liu et al employ a similar GT inspired mechanism in a clonal selection algorithm, which is shown to provide improved performance in automatic clustering problem (Liu et al, 2009). In a related research, McGregor and Harvey use a mechanism similar to transposition which they termed as “plagiarism” (McGregor and Harvey, 2005). The “plagiarism” copies one part of the genotype into another, replacing the latter completely. The authors demonstrated that the proposed mechanism improves the performance of the evolution of solutions to the Boolean logic problems. Spirov et al. also develop an original implementation of artificial transposition, used as a form of mutation operator for the simulated evolution of evolving a finite state machine as a solver of the artificial ant problem (Spirov et al. 2009).

In these aforementioned works, as well as in the biology, GT can occur frequently during the evolutionary cycle (just like other common evolutionary operations, such as crossover). In the approach we propose, however, GT occurs only once for each “seeding phase” (which, in turn, is only once per evolutionary run – at the stage of creating the initial population), and not invoked during the evolutionary run. Therefore, although the source of inspiration is the same, the implementation of the proposed model differs significantly from the previously developed GT-inspired mechanisms in EC. However, for the rest of the paper, we will refer to the GT-inspired mechanism introduced here as genetic transposition (GT) for simplicity and succinctness.

In our work we are especially interested in achieving higher efficiency in GP for coevolution of locomotion gaits and sensing of the simulated Snakebot. At the initial stage of the proposed approach, we evolve a pool of generic fast-moving sidewinding bots in a flat, smooth terrain. Then, during the second stage, we use these Snakebots to seed the initial population of the bots that are further subjected to coevolution of their locomotion control and sensing in a more challenging environment. During the seeding process the generic, fast moving, sensorless bots are subjected to genetic retro-transposition (i.e., duplicated within the same genome). The resulting transposon (connected with the seeding genome via a randomly initialized “control gene”) is subjected to 100% random mutation in order to allow for the incorporation of the sensing information into the locomotion control of the bot. The schematic parse tree of the genotype of Snakebot, created during the initialization of GP via the GT-inspired mechanism is illustrated in Figure 1.

Seeding of the initial population by means of including the previously evolved successful (or partially successful) solutions has been shown to be an effective way of improving the efficiency of simulated evolution. For example, Nolfi et al. (Nolfi et al. 1994) evolve the controller of simulated robot and then re-evolve (or, adapt) the obtained results on real robots to accelerate the evolutionary process. Other examples of successful seeding include the work of Vassilev et al. (Vassilev et al. 2000) on the optimization of the existing digital circuit design; Thomsen et al. (Thomsen et al. 2002) on the use of solution obtained from a domain-neutral algorithm as a seed to evolve an even better performing solution; Langdon et al. (Langdon and Nordin, 2000) on seeding the evolutionary population with hand-coded solutions that allow a better generality of the evolved results.

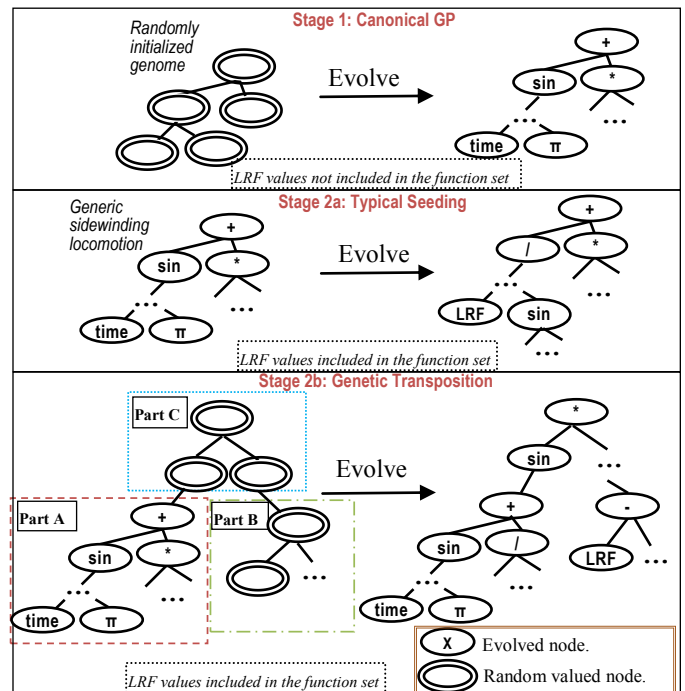


Figure 1: The mechanism of proposed genetic transposition in GP (Stage 2b) and the typical seeding process (Stage 2a). Both of these cases need to make use of a preliminary seed, and in the proposed approach this seed comes from a previously evolved sensorless Snakebot (Stage 1) that achieves fast locomotion on a smooth open terrain. In either of the Stages 2a and 2b, the resulting genome from Stage 1 is used as a seeding individual and further evolved, with additional sensory abilities (illustrated by the terminal symbol LRF) in a more challenging terrain. For Stage 2a the seed from Stage 1 makes up the whole genome of the initial Snakebot. For Stage 2b, the seed from Stage 1 is only a part (Part A) of the initial genome of the Snakebot. The rest of it contains a clone of the seed that has gone 100% mutation (Part B), and a randomly initialized group of control gene (Part C) which connects Parts A and B.

In addition, by utilizing the previously evolved solutions, seeding has also been applied to improve the performance of evolution of solutions from scratch. This technique, termed by Perry as “population enrichment” (Perry, 1994), has been demonstrated to be more efficient in discovering solutions in GP. “Population enrichment” is a form of seeding that is closest to the GT technique described in this paper. The main difference in these methods is the form of initialization, where in the “population enrichment” the seed is used to create the complete individual (see Stage 2a in Figure 1), while in GT the seeded genotype only forms a part of the genetic makeup of the newly created individual in the initial evolutionary population (see Stage 2b in Figure 1).

## Evolutionary Framework and The Simulation Environment

In the experiments presented in this work we employed a DOM/XML-based implementation of GP (Tanev, 2004). The

benefits of representing the genetic programs as DOM-parse tree featuring text-based XML-representation of genetic programs are (i) fast prototyping of GP by using standard built-in API of DOM-parsers for traversing and manipulating genetic programs, (ii) generic support for the representation of grammar of strongly-typed GP using W3C-standardized XML-schema; and (iii) human-friendly, text-based representation of the evolved solutions.

## Representation of the Snakebot

We employ open dynamics engine (ODE) as a simulation platform for the Snakebot. ODE is a free, industrial quality software library for simulating articulated rigid body dynamics (Smith, 2004). It is fast, flexible and robust, and it has built-in collision detection. Therefore, ODE is suitable for a realistic simulation of the physics of an entire Snakebot when applying actuating forces to its segments. The ODE related parameters of the simulated Snakebot are same as elaborated in (Tanev et al. 2005).

Snakebot is simulated in ODE as a set of 15 identical spherical morphological segments ("vertebrae"), linked together via universal (Cardan) joints (Figure 2). All joints feature identical angle limits and each joint has two attached actuators ("muscles"). A single LRF sensor, with a limited range is rigidly attached to each of the segments.

The functionality of the LRF can be defined by the values of the following set of parameters: (i) orientation, measured as an angle between the longitudinal axis of the sensor and the horizontal axis of the joint, (ii) range of the sensor (in cm), and (iii) the timing of activation, expressed as a threshold value of the turning angle of the horizontal actuator. The reading of LRF is a scalar value which corresponds inversely to the distance between the sensor and an object (if any within the sensor's range), measured along the longitudinal axis of the LRF. In the initial standstill position of Snakebot the rotation axes of the actuators are oriented vertically (vertical actuator) and horizontally (horizontal actuator) and perform rotation of the joint in the horizontal and vertical planes respectively.

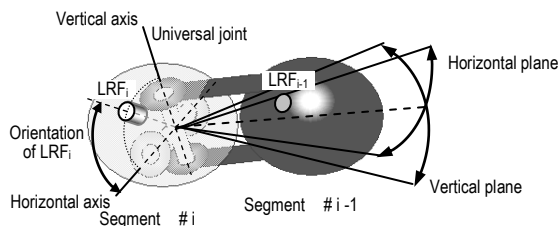


Figure 2: Horizontal and vertical actuators attached to the joint perform rotation of the segment #i-1 in vertical and horizontal planes respectively.

Considering the representation of Snakebot, the task of designing the fastest locomotion can be rephrased as developing temporal patterns of desired turning angles of horizontal and vertical actuators of each segment that result in fastest overall locomotion of Snakebot. The proposed representation of Snakebot as a homogeneous system comprising identical morphological segments is intended to

significantly reduce the size of the search space of the GP. Since the size of the search space does not necessarily increase with the number of morphological segments of the Snakebot, the proposed approach offers a favorable scalability.

## Algorithmic Paradigm

For the evolution of the Snakebot, the genotype is represented as a triple consisting of a linear chromosome containing the encoded values of the three relevant parameters of LRF, and two parse trees corresponding to the algebraic expressions of the temporal patterns of the desired turning angles of both the horizontal and vertical actuators, respectively (Figure 3).

The encoding of the parameters of LRF is as elaborated in Figure 3. The same figure also illustrates the function set and the terminal set of the GP, employed to evolve the control sequences of both actuators. Because the locomotion gaits by definition are periodical, the periodic functions sine and cosine are included in the function set of GP in addition to the basic algebraic functions. Terminal symbols include the variables time, segment\_ID, an automatically-defined function (ADF), the reading of the sensor (LRF), and two constants: Pi, and a random constant within the range [0, 2]. The incorporation of the terminal symbol segment\_ID (a unique index of morphological segments of Snakebot) allows GP to discover how to specialize (by phase, amplitude, frequency etc.) the

genetically identical motion patterns of actuators of each of the morphological segments of the Snakebot.

The rationale of employing ADFs is based on the observation that the evolvability of straightforward, independent encoding of desired turning angles of both horizontal and vertical actuators is rather poor. Even without ADFs, GP is able to adequately explore the potentially large search space and ultimately discover the areas that correspond to fast locomotion gaits in the solution space. However, it was observed in the previous work of Tanev et al (Tanev et al. 2005) that not only the motion patterns of adjacent segments are correlated, but the motion patterns of horizontal and vertical actuators of each segment in fast locomotion gaits are highly correlated too. Moreover, discovering and preserving such correlation by GP is associated with enormous computational effort. ADFs, which provide a way of introducing modularity and reuse of code in GP (Koza, 1994), are employed in our approach to allow GP to explicitly evolve the correlation between motion patterns of horizontal and vertical actuators as shared fragments in algebraic expressions of desired turning angles of both actuators. Furthermore, we observed that the best results are obtained by; (i) allowing the use of ADF as a terminal symbol in algebraic expression of desired turning angle of vertical actuator only, and (ii) evaluating the value of ADF by equalizing it to the value of currently evaluated algebraic expression of desired turning angle of horizontal actuator. The main GP (hence the EA) parameters are summarized in Table 1.

**Genetic Operations.** We employ a binary tournament selection and a single point crossover. The crossover point is randomly selected between the three components of the genotype (as shown in Figure 3). The mutation randomly alters either a value of an allele in the linear chromosome

representing the parameters of LRF, or a sub-tree in one of the two parse trees that correspond to the temporal patterns of the control sequences of actuators.

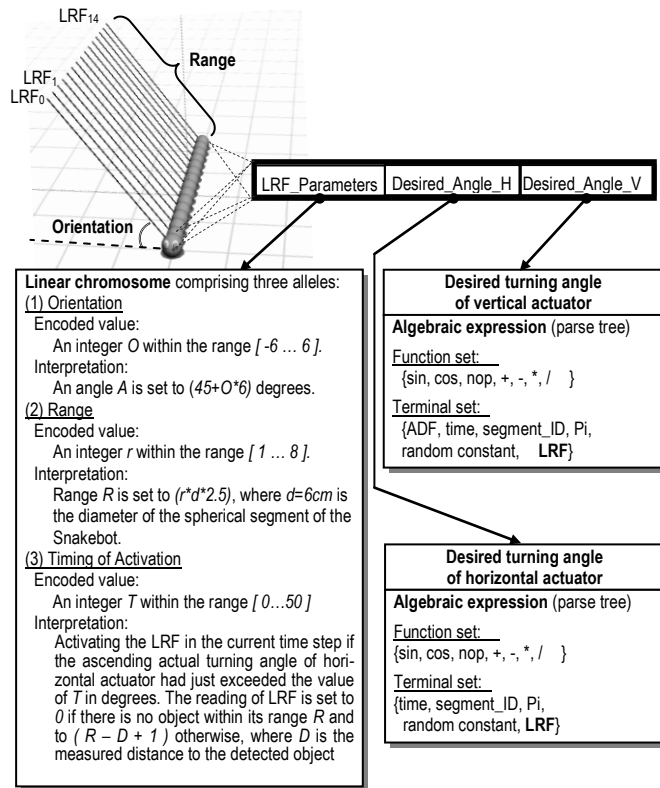


Figure 3: Genotype of the Snakebot, represented as a triple containing the values of the parameters of LRF and two algebraic expressions of the temporal patterns of the desired turning angles of horizontal and vertical actuators, respectively.

**Fitness Evaluation.** The fitness function is based on the average velocity of Snakebot, which is estimated from the distance traveled during the trial. As we shall elaborate later in the “Experimental Setup” section, the confined environment used in the trial is a narrow corridor covered with obstacles of various sizes (Figure 4). The velocity of locomotion needed to clear the final obstacles towards the end of the corridor for the given time of the trial (16s) corresponds to a fitness value of 100. The evolution is terminated if the bot reaches the fitness of more than 120 (fitness required to clear the whole corridor) or if the maximum accumulative number of 80 generations is reached. 80 generations was set as the cumulative maximum as a result of the experience from earlier experiments. Earlier experiments used in achieving locomotion of modular Snakebot had used 40 generations, which was a sufficient limit for the evolution of locomotion. Ideally, addition of a new feature should not require a much larger computational effort. Therefore, 40 generations per individual feature of the Snakebot was decided to be an acceptable cost.

**Experimental Cases.** In order to investigate comparatively the efficiency of proposed approach, we used three methods to evolve the locomotion of Snakebot with sensors:

Category	Value
Population Size	200
Selection	Binary Selection ratio: 0.1 Reproduction ratio: 0.9
Elitism	4
Mutation Rate	1%
Trial Interval	16s (400 time steps of 40 ms per step)
Termination Criterion	(Fitness=120) or (Num. of Generations=80)

Table 1: The GP-related parameters.

- I. *Canonical GP* (single stage approach): In this case the evolution of the Snakebot is done from scratch; i.e. evolution starts with a population of randomly created individuals and optimizes these individuals to satisfy the target fitness. The limit of the evolutionary generations of GP is set to 80.
- II. *Typical seeding* (two-staged approach): The genotypes of six best sensorless Snakebots that have already been evolved to achieve fast sidewinding locomotion in a plain, smooth terrain (Figure 1, Stage 1), is used to create the initial population. This evolved genotype is used as an elite individual to seed the initial population, where the exact copies of these six sensorless bots are used to form a small part (6 bots) of the initial evolutionary population. The remaining part of the population (194 bots) is randomly generated. This seeded population is then evolved to fully satisfy the target fitness (Figure 1, Stage 2a). The limit of the generations of both stages of evolution is set to 40.
- III. *GT* (two-staged approach): The first stage of the proposed approach is identical to that of the typical seeding method. Similarly, the six best sensorless genotypes are used as elite individuals in the initial population of the second evolutionary stage. To create the remaining 194 bots of the initial population, however, we use the evolved best sensorless genotypes to form only part of these newly created individuals. The remaining parts of these individuals are created randomly, as elaborated in the section titled “Genetic Transposition in GP”. These 194 partially seeded and partially random individuals and the six fully seeded individuals are used to form the initial evolutionary population (Figure 1, Stage 2b). The created population is evolved to fully satisfy the target fitness. Similar to the typical seeding, the limit of the generations of both stages of evolution is set to 40.

## Experimental Setup

The experimental environment (Figure 4a and 4b) is formed of a straight narrow corridor (the width is the same as the length of the Snakebot) that has two groups of tall boxes that protrude to about 40% of the width of the corridor. In addition, part of the corridor is covered by many, randomly located and sized, small boxes that are designed to create a



rough terrain and noisy environment for the sensors. The length of the corridor is set to seven times the length of the Snakebot. Starting from one end of the corridor, the aim of the bot is to reach the other end within the given time-span. We designed this environment with the intention to encourage the evolving bot to develop the following abilities: (i) fast locomotion (long enough corridor), that is (ii) not hindered by rugged terrain (small boxes), (iii) following of obstacles that cannot be overcome (walls), and (iv) circumnavigating obstacles that cannot be overcome (tall boxes).

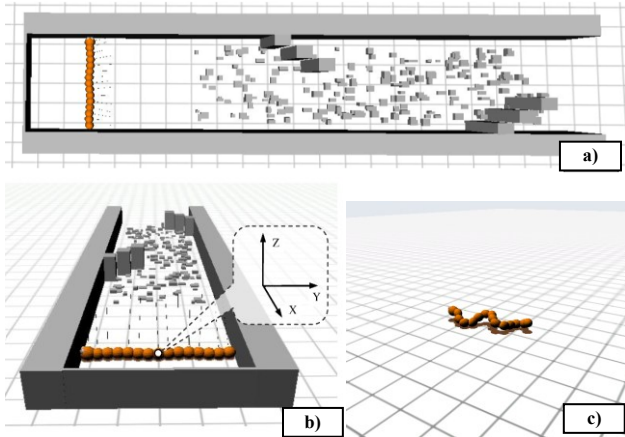


Figure 4: The experimental setup of the scenes.

For experimental case I and Stage 2 of experimental cases II and III, the Snakebot is initialized with 15 modules, on full stretch at the dead end of the corridor with its longitudinal axis perpendicular to the intended direction of movement (Figure 4a, and 4b). Initially, the rough terrain is not present to facilitate the evolution of basic locomotion on smooth terrain. After a fitness value of 60 is reached (i.e. the first set of large obstacles can be cleared by the Snakebot), a large portion of the corridor is filled with randomly initialized boxes (random size and location). The initial orientation of the Snakebot and the corridor is influenced by the previous work suggesting that sidewinding is the fastest and most robust locomotion gait for a Snakebot. Therefore, the Snakebot would be expected to enter a corridor featuring a similar orientation.

For Stage 1 of experimental cases II and III, a plain surface with no obstacles is used as the environment (Figure 4c), and the LRF is excluded from the GP function-set.

## Experimental Results

The Snakebot is evolved applying the three different evolutionary approaches as described in the “Experimental Cases” section, and under the experimental conditions as outlined in the “Experimental Setup” section. For each approach we executed 38 independent runs. The fitness convergence characteristics of these runs are shown in Figure 5. As Figure 5a depicts, the canonical GP features average fitness (over all independent runs) of about 40, which corresponds to the 40% of the length of the corridor, which also corresponds with the position of the first set of tall

obstacles. The pace of the improvement of the fitness is rather slow, with average value of about 30 at generation 40. These results suggest that the bot is struggling to discover the generic locomotion gaits that can result in a fast enough locomotion even in the absence of obstacles. The large search space of the evolution, caused by the need to additionally evolve the sophisticated morphology sensing (LRF), and the way to properly incorporate the sensing signals into the locomotion control is one of the reasons for the poor efficiency. Another reason is the challenging environment—the walls and various obstacles, which implies that the fitness landscape of evolution features fewer (compared to the previously tested cases [Tanev et al. 2005; Tanev and Shimohara, 2008]) and narrower optimal areas. Indeed, even if fast locomotion emerges during the initial stages of evolution, its survival value could be easily “underestimated” by evolution because the bot gets stuck at the first obstacle. Hence the large difference between the progression of the results displayed in Figures 5a and 5b.

	Canonical	Seeding	GT
Average Fitness	43	69.3	91.2
Median Fitness	37	67	91
Std Dev. of Avg. Fitness	23.5	27.2	19.4
Runs with Fitness >100	1 (2.6%)	3 (7.9%)	8 (21%)

Table 2: Statistics of the experimental results.

Conversely, the results of the first stage of both the typical seeding and GT (Figure 5b) indicate that the evolution of the locomotion of a Snakebot is more efficient, when relieved from the burden of dealing with the sensors and the sophisticated environment. The velocity of 100, which would be sufficient to clear the obstacles, is now easily achievable within 10 to 36 generations.

Then, when six of these best moving generic bots are incorporated via typical seeding into the initial population in the second stage of evolution (Figure 5c), and allowed to evolve for additional 40 generations, the average fitness value is 1.6 times higher than the result obtained by canonical GP (Table 2). However, the best efficiency of evolution is achieved when GT is used—the average fitness is more than 90 with 8 successful runs, and a smaller deviation in the fitness values achieved (Figure 5d and Table 2).

The proposed approach of employing GT allows the evolution to experiment with the way of processing the sensory signals without the risk of damaging the already evolved, fast locomotion control. Therefore, the transposition could facilitate the protection of the already evolved beneficial building blocks from the destructive effects of genetic operations. Conversely, since the locomotion control comprises 100% of the genotype of the bots created via typical seeding, any incorporation of the sensing information as a result of the genetic operation would most likely result in damage of this control. Indeed, the genotypes of the successful results achieved via GP with GT, the resulting genotype always had a portion in the form of Equation 1. Equation 1 (with C1 and C2 being constants) is the general form of the controllers achieved via the evolution experiments for the locomotion of a Snakebot on a smooth, empty terrain,



i.e. Figure 5b. The resulting genotypes were simply the modulation of these controllers via the LRF signals. On the other hand the successful solution achieved by canonical GP and typical seeding runs did not have an exclusive part of their genotype that resembles Equation 1; instead a large, complicated equation that is hard to comprehend was the result.

$$C1 * \sin(ID + time + C2) \quad (1)$$

In fact, when re-run on maps with differently arranged obstacles (to that of the environment present during evolution), the most robust Snakebots are observed to be from the GP runs using GT. We believe that the following are the reasons for the significant improvement in the efficiency (computational effort) of evolution due to GT:

- A wider spread of the initial seed into the population (than the typical seeding) of genotype that features generic ability to move,
- A better value of the initial fitness of the bots as they already feature the generic ability to move in their genotypes, and
- A separation of the sensing and locomotion parts of the genome, which may create a more efficient control mechanism for the bot.

We would like to point out that latter of the above mentioned arguments might provide a further insight into the design of robotic control systems and their sensorimotor control. The locomotion property of the Snakebot can be viewed as a continuous process that needs to be applied regularly under normal conditions, and the sensing property of the Snakebot can be viewed as a reflex that only needs to affect the actions of the bot when an event occurs. Such a concept might be seen as analogous to the reactive behavior related to the reflexes observed in biological organisms. For example, the collision-free flight of locusts in a crowded swarm is recognized to be achieved by direct, real-time input of the sensory signals into the wings muscles. The latter serve as a mediator for both the (i) “default” oscillating signals (generated by CPG) and (ii) the visual sensors (Uvarov, 1977).

From another viewpoint, our results can be seen as an evidence of the computational benefits of mimicking the neurobiological concept of achieving complex navigation behaviors of species in nature through sensory-controlled modulation of CPG. The moving trajectory of a sample best of run bot (Figure 6) illustrates the emergence of the following abilities of the bot: (i) fast locomotion (clearing the corridor), that is (ii) not hindered by rugged terrain (overcoming small boxes), (iii) following obstacles that cannot be overcome (walls), and (iv) circumnavigating obstacles that cannot be overcome (two groups of tall boxes).

The successful Snakebots from the results presented demonstrate the incorporation of sensor information within the control mechanism of the Snakebots for steering the Snakebot away from the large obstacles. The evolved Snakebots use the sensor signals as repulsive forces on the individual modules, which gradually change the course of the whole Snakebot.

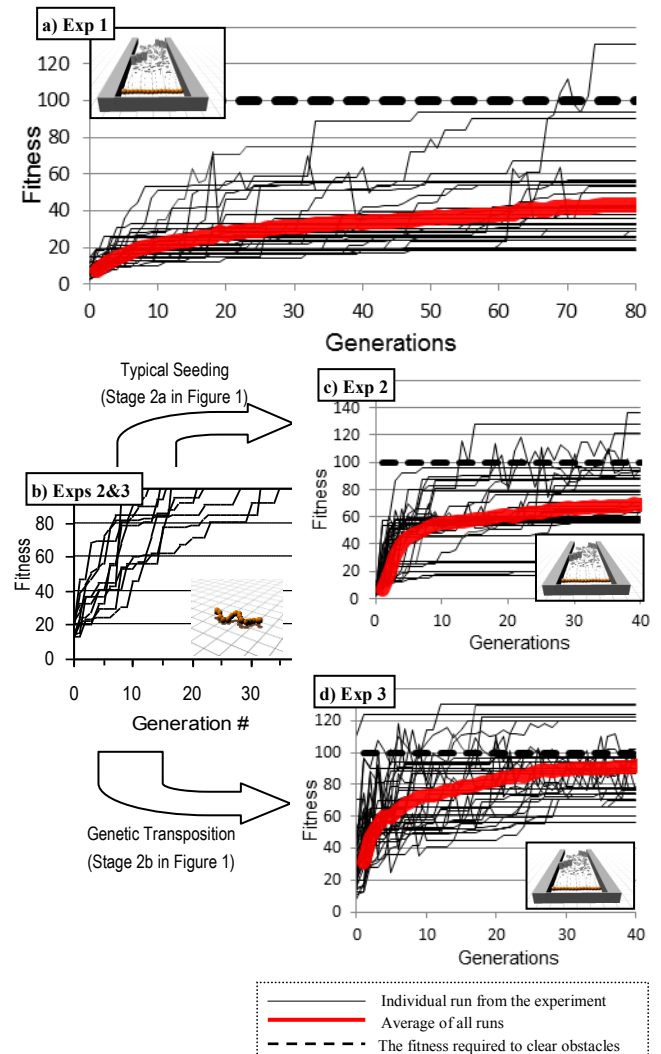


Figure 5: Fitness convergence characteristics of the three approaches used to evolve the Snakebot in the confined environment: single-staged canonical GP (a), and incremental two-staged typical seeding (b then c) and GT (b then d), respectively. The graphs show the fitness convergence of all 38 runs from each experiment.

## Conclusions

We demonstrated that the evolution of a modular sidewinding Snakebot in a challenging environment with multiple forms of obstacles is a computationally demanding task. Dividing this task into two subtasks, implemented as two consecutive evolutionary stages, contributes to the significant improvement in the efficiency of evolution.

We introduced a genetic transposition inspired seeding technique to further improve both the quality of the bots and the computational effort required to evolve them. The proposed technique offers a significant improvement over typical seeding when applied to the evolution of an active sensing of fast moving Snakebot. The presented technique could be seen as a promising approach to incremental coevolution of multiple features of morphologically and

behaviourally complex bots situated in challenging environments.

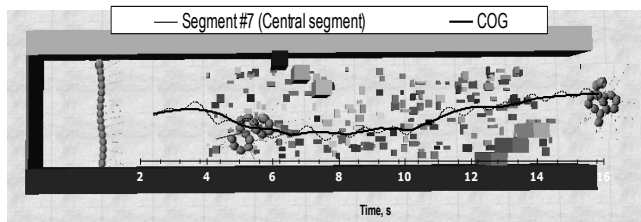


Figure 6: The moving trajectory of the central segment and the center of gravity (COG) of a sample best-of run Snakebot, evolved by incremental GP with GT.

In biology it is recognized that the transposons (the jumping genes) facilitate the evolution of increasingly complex forms of life by providing the creative playground for the mutation where the latter could experiment with developing novel genetic structures without the risk of damaging the already existing, well-functioning genome. The results shown in this paper demonstrate that this biological occurrence is also applicable to EC, and the proposed genetic transposition inspired seeding mechanism also facilitates the artificial evolution of increasingly complex systems.

As part of future work, we aim to analyze the Snakebots evolved in detail in order to gain an understanding of how the sensory signals are integrated with locomotion, and to infer the definition (understandable by human designers) of the control mechanism of the Snakebot. Furthermore, we plan on studying and designing mechanisms that can accompany genetic transposition in bringing a more efficient and robust evolution of complex robotic systems with multiple evolving features. Finally, we aim to generalize the proposed technique and define the properties of the tasks in evolutionary robotics that can be efficiently solved via this approach.

## References

- Chan, T.; Man, K.; Tang, K. & Kwong, S. (2008). A Jumping Gene Paradigm for Evolutionary Multiobjective Optimization. *Evolutionary Computation*, IEEE Transactions on, 12, 143-159
- Dowling, K. (1999) Limbless locomotion: learning to crawl In *Robotics and Automation*. Proceedings. 1999 IEEE International Conference on, volume 4, pages 3001–3006 vol.4, 1999
- Hirose, S. (1993) *Biologically Inspired Robots: Snake-like Locomotors and Manipulators*. Oxford University Press.
- Koza, J. R. (1994) *Genetic programming II: automatic discovery of reusable programs*. MIT Press, Cambridge, MA, USA.
- Koza, J. R., Keane, M.A., Yu J., Bennett, F.H., and Mydlowec W. (2000) Automatic creation of human-competitive programs and controllers by means of genetic programming. *Genetic Programming and Evolvable Machines*, 1:121–164. 10.1023/A: 1010076532029.
- Langdon, W.B. and Nordin, P. (2000) Seeding genetic programming populations. In *Proceedings of the European Conference on Genetic Programming*, pages 304–315, London, UK. Springer-Verlag.
- Levitin, I.B. and Kaczmarek, L.K. (2002). *The Neuron—Cell and Molecular Biology*. Oxford University Press.
- Liu, R.; Sheng, Z. & Jiao, L. (2009). Gene Transposon Based Clonal Selection Algorithm for Clustering. *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, 1251-1258.
- McClintock, B. (1950) The origin and behavior of mutable loci in maize. *Proc Natl. Acad. Sci. USA*, 36:344–355.
- McGregor, S. and Harvey, I. (2005). Embracing plagiarism: Theoretical, biological and empirical justification for copy operators in genetic optimisation. *Genetic Programming and Evolvable Machines*, 6:407–420.
- Morowitz, H.J. (2002). *The Emergence of Everything: How the World Became Complex*. Oxford University Press.
- Nolfi, S., Floreano, D., Miglino, O. and Mondada, F. (1994). How to Evolve Autonomous Robots: Different Approaches in Evolutionary Robotics. In *4<sup>th</sup> International Workshop on Artificial Life*. MA: MIT Press, 1994.
- Nowacki, M., Higgins, B.P., Maquilan, G.M., Swart, E.C., Doak, T.G., and Landweber, L.F. (2009). A functional role for transposases in a large eukaryotic genome. *Science*, 324 (5929): 935–938.
- Perry, J. (1994). The effect of population enrichment in genetic programming. In *Evolutionary Computation. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pages 456–461 vol.1, June 1994.
- Simões, A. and Costa, E. (2000). Using genetic algorithms with asexual transposition. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'00)*, pages 323–330. Morgan Kaufmann.
- Simões, A. and Costa, E. (1999). Transposition: A biologically inspired mechanism to use with genetic algorithms. In the *Proceedings of the Fourth International Conference on Neural Networks and Genetic Algorithms (ICANNGA'99)*, pages 612–619. Springer-Verlag.
- Smith, R. (2004). *Open Dynamics Engine*.
- Spirov, A.V., Kazansky, A.B., Zamdberg, L., Merelo, J.J., and Levchenko, V.F. (2009). Forced evolution in silico by artificial transposons and their genetic operators: The John Muir Ant Problem. Technical Report, Oct 2009.
- Strand, D.J. and McDonald, J.F. (1985). Copia is transcriptionally responsive to environmental stress. *Nucleic Acids Research*, 13(12):4401–4410.
- Tanev, I., Ray, T., and Buller, A. (2005). Automated evolutionary design, robustness and adaptation of sidewinding locomotion of simulated snake-like robot. *IEEE Transactions on Robotics*, 21:632–645.
- Tanev, I. and Shimohara, K. (2008). Coevolution of active sensing and locomotion gaits of simulated snake-like robot. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation, GECCO '08*, pages 257–264, New York, NY, USA. ACM.
- Tanev, I. (2004). Dom/xml-based portable genetic representation of the morphology, behavior and communication abilities of evolvable agents. *Artificial Life and Robotics*, 8:52–56.
- Thomsen, R., Fogel, G., and Krink, T. (2002). A clustal alignment improver using evolutionary algorithms. In *Evolutionary Computation. CEC '02. Proceedings of the 2002 Congress on*, volume 1, pages 121–126, May 2002.
- Uvarov, B. (1977). *Grasshoppers and Locusts*, volume 2. 1977.
- Vassilev, V.K., Job, D., and Miller, J.F. (2000). Towards the automatic design of more efficient digital circuits. In *EH'00: Proceedings of the 2nd NASA/DoD workshop on Evolvable Hardware*, page 151, Washington, DC, USA. IEEE Computer Society.

# Multi-Agent Systems and Heterogeneous Scales Interactions. Application to Pharmacokinetics of Vitamin K Antagonists

Sébastien Le Yaouanq<sup>1</sup>, Pascal Redou<sup>1</sup>, Christophe Le Gal<sup>2</sup>,  
Jean-François Abgrall<sup>3</sup> and Jacques Tisseau<sup>1</sup>

<sup>1</sup>European Center for Virtual Reality, LISyC EA3883

<sup>2</sup>CERVVAL, 29280 Brest, France

<sup>3</sup>Hematology laboratory, Brest University Hospital, France  
leyaouanq@cervval.com

## Abstract

The study of complex systems consists in considering entities submitted to interactions which define the dynamics of the system. Virtual reality opens the way to interactive simulation of complex systems, so called the *in virtuo* experimentation. For that purpose we use multi-interactions systems, based on the reification of interactions and multi-agent systems, in a phenomenological approach. Interaction agents represent the modeler understanding of the relations between the constituents of the system. Such descriptive models lead us to define parameters *a priori*. Moreover these parameters can be fluctuant, or even unknown, during a simulation in relation to the system dynamics or user interventions. To respond to this problem, we expose in this paper a redundant multiscale architecture which rests upon the fact that we can establish models of a same phenomenon at heterogeneous time and space scales. Heterogenous Multiscale Methods provide a general framework to mix levels of description of a system. Our intention is to implement this framework in multi-interactions systems by means of a *Scale-Interaction* agent. Then we illustrate our architecture through a pharmacokinetics application. Indeed biochemical kinetics abounds of parametric phenomena. Finally we discuss about some questions raised by this methodology, such as synchronicity, organization detection and genericity.

## Introduction

The study of complex systems consists in considering entities in interaction. These interactions affect the entities behaviors and then change the system dynamics. The entities are most often heterogeneous by their natures, their interactions and their scales. Moreover, their great number is a major obstacle to their understanding. Thus we have to model these systems, even roughly, in order to get out some new knowledge. It's generally difficult formally to prove that a model is exact. That's why we need to experiment our models so as to compare simulations and observations.

Virtual reality enables us to manipulate these models (Fuchs et al., 2006). An expert can be immersed in real time within a virtual laboratory, mock up the system he wants to study and experiment it, without any danger or consequences. This is called the *in virtuo* experimentation by analogy with *in vivo* and *in vitro* methods. It allows

the modeler to build his model incrementally, by successive additions of phenomena. This is a "phenomenological approach" for modeling (Parenthoën, 2004).

For that purpose we use multi-interactions systems (MIS), based on the reification of interactions and multi-agent systems. It consists in changing our point of view to describe phenomena just as we observe them. Thus agents are not the entities anymore but the interactions binding them. This method has been repeatedly successfully applied, validated (Redou et al., 2007) and today, there are several models and methodologies that can be used to experiment complex systems with multi-agent systems (Desmeulles et al., 2009).

In addition, this kind of modeling has the advantage of reducing the computation time because phenomena are described macroscopically with the help of ordinary differential equations (ODE). The price is that we often have to define parameters in models, like diffusion coefficients for instance. Moreover these parameters can fluctuate, or even unknown, during a simulation in relation to the system dynamics or user interventions (Béal et al., 2010).

To respond to this problem, we propose in this article to make maximum use of the knowledge we have about the phenomena. We expose then a redundant multiscale architecture which rests upon the fact that we can establish models of the same phenomenon at heterogeneous time and space scales. Parameters of a macroscopic model are in fact related to the system dynamics at microscopic scale. For instance, diffusion rate of a chemical concentration can be determined using brownian motion of molecules and statistical physics (Frenkel and Smit, 2001).

Therefore, our idea is to run parallel simulations of multiple description scales in order to parameterize phenomena. Interactions between the scales will be supported by agents, which is the core of this article.

We will illustrate our architecture through a pharmacokinetics / pharmacodynamics (PK / PD) model of the vitamin K antagonists (VKA). Chemical kinetics is indeed a perfect example of parametric phenomena. For this occasion we will discuss about some questions raised by our method.

## Heterogeneous multiscales modeling

In the perspective of a phenomenological approach, we are interested in the effects of the phenomena over entities. These effects are usually described by differential equations at a macroscopic scale. This approach is very successful for large classes of problems but it favours efficiency over accuracy introducing empirical closures and parameters in equations that are often partially known or understood. Besides, the system dynamics can be unpredictable. So an acceptable closure or parameter at a given instant could become wrong at the next one. In the case of more complex systems it seems to be necessary to call upon different methods, particularly by coupling models with different levels of description in order to achieve a balance between accuracy and efficiency. We talk then about multiscale methods.

Multiscale methods have been existing for a long time, such as adaptative mesh refinement methods (Debreu et al., 2008). Their purpose is to mix different scales, solved separately, into a global simulation. Such an idea can be applied in the context of stiff ODEs resolution using splitting methods (Le Bris, 2005; Guibert, 2009).

Let us consider the example of a system  $z$  composed of  $C$  operators where  $A$  (resp.  $B$ ) are non-stiff (resp. stiff) operators :

$$\frac{dz}{dt} = Cz = Az + Bz \quad (1)$$

We can solve  $z$  over each time step  $[n\Delta t, (n+1)\Delta t]$  with

$$\begin{cases} \frac{dz^{**}}{dt} &= Bz^{**} \\ z^{**}(n\Delta t) &= z^*((n+1)\Delta t) \end{cases} \quad (2)$$

A different solver could be used for each part of the system in this way, potentially with a smaller time step for the stiff one.

Classical multiscale methods are extremely accurate but their cost can be huge. Indeed, their efficiency is closely dependent on the smaller time step used in the simulation.

That's why recently developed multiscale methods aim at one step further : in order to cut down the computing time, they try to capture the macroscale behavior of the system from local microscale simulations run over a limited time (Horstemeyer, 2009).

Heterogeneous multiscale method (HMM) (Weinan et al., 2007) relies on the following concept : coupling redundant scales in order to take into consideration possible variations of the system. In some cases, the macroscopic model is not explicitly available or is invalid in some part of the domain. The microscopic model is used then to supply the necessary data for the macroscopic model. Scale separation is exploited so that coarse-grained variables can be evolved on macroscopic scale using data that are predicted based on the simulation of the microscale ; see figure 1.



Figure 1: Schematics of HMM framework (Weinan et al., 2007)

We consider a macroscale with a state variable  $U$ . We have seen that this state depends on the system dynamics and / or parameters. We have at our disposal a microscopic model, such as molecular dynamics, that describes the microscopic state variable  $u$  of the same system. Let's not forget that we are dealing with different levels of description of an unique system.

The two scales are related one to each other by the use of reconstruction ( $Q$ ) and compression ( $R$ ) operators :

$$\begin{cases} Q.u &= U \\ R.U &= u \end{cases} \quad (3)$$

with the property  $Q.R = I$ , where  $I$  is the identity operator. The role of these operators is to translate the system structure and dynamics from a scale to the other. The main difficulty lies in the definition of these operators. Indeed it would be naïve to think that a macroscale phenomenon could be the result of an unique microscopic one. Most often we would be interested in a group of local interactions from which emerge a global behavior. This underlines one of primary interests of multiscale and complex systems simulation : we may increase our understanding of phenomena by observing their entanglements (Lesne, 2003).

Finally the idea is to make round-trips between the two scales regarding to the system dynamics. As soon as there is a lack of data in the macroscale ensued from a dynamics variation, we use  $Q$  to rebuild a microscale which we run and observe over a given duration. Thus we can make data estimations with the help of  $R$  so as to set new parameters in the macroscale.

HMM give guidelines on how to design redundant multiscale systems. It is a general framework which is lacking implementation. We intends in this paper to fill this lack by the use of multi-agent systems.

We saw that HMM deduce the macroscale behavior according to emergent processes and data from the microscale. Concept of emergence is one basis of multi-agent systems (Demazeau, 1995). Into such systems, autonomous entities evolve with only a partial knowledge

of their environment. The addition of their individual interactions results in a collective behavior. Then multi-agent systems appear as very good tool to model complex systems by the way of splitting the whole problem in smaller subparts. Today, it has been successfully applied to molecular dynamics for instance (Parisey, 2007). However, it implies to simulate each entity separately which is a very costly method. That's why we can say that classical multi-agent systems match much more to a microscale method.

As we have seen before, multi-interactions systems offer an original approach to model efficiently the links between entities. We can also discern that HMM operators could be considered as links between scales. That's why we propose in the following section to implement this process into multi-interactions framework which we use to perform *in virtuo* experiments.

### Virtuo framework : MIS implementing HMM

Multi-interactions systems were first presented in the RéISCOP meta-model (Desmeulles, 2006). The motivation was to enable an expert to describe a system as he observes it in the nature, usually with the help of ODE. Subsequently, we pursued this work bringing the reification of numerical methods of solving used to make the system evolve during the simulation (Le Yaouanq, 2010). This allows us to keep control over the convergence and stability of the system. This modification and others prompted us to propose a new framework that implements MIS : we called it Virtuo.

Our will is now to take the advantage of redundant multiscale methods in order to parameterize macroscopic descriptions of phenomena. *In virtuo* experimentation puts interactivity first. It implies to consider some constraints such as real-time computing and reactivity of the simulation. In this context, we can not use classical multiscales methods which impose the choice of small time steps. That is where we join the HMM framework. Our idea is to simulate critical phenomena at a microscale selectively and for a limited duration so as to complete their description.

In this section, we detail Virtuo architecture and how it makes possible to model multiscale systems taking inspiration from HMM.

### How to design the model of a given scale ?

As said above, we are interested in systems composed of numerous entities. They are represented by the *Entity* class in our model ; see figure 2. To fit reality, we introduce a concept of hierarchy between entities. Thus an entity can contain other entities. It is important to note that we are still talking about a same level of description. This ability is just a way of considering spatial organizations into a system. For instance, from a macroscopical point of view, a human body is composed of organs. But it doesn't mean that there are two different levels of description. The decisive element that will encourage us to consider multiple scales into a

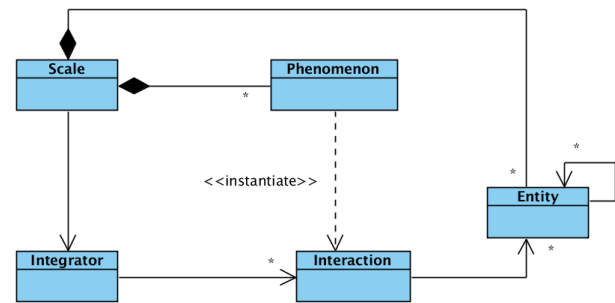


Figure 2: Class diagram focusing on Scale designing (Le Yaouanq, 2010)

simulation is the time / spatial reach of interactions. This will be discussed thereafter.

Multi-interactions systems argue in favour of considering that active agents in the system are the interactions between passive entities. So an *Interaction* agent associates one or many entities and compute their local effects on each other. The systems's dynamics is the addition of these local modifications.

An *Interaction* computing consists in solving independently a part of an ODE system. For this we use numerical methods to make the system evolve on each time step (Ascher and Petzold, 1998). Even if they have been validated in the context of multi-interactions systems, they force us to take care of the system convergence and stability (Redou et al., 2010). Actually, if the time step is chosen too large, some interactions could induce an irreversible instability of the whole system. That is why we add the *Integrator* agent whose job is to manage the interactions. It would be able to control *Interactions* actions and order them to recompute more precisely if needed.

The desynchronization of interactions eases a modular and incremental building of the numerical model. Firstly, this is especially useful for online models building, since the modeller usually selects, subjectively, the phenomena that are most likely involved, and runs the model. If results are not correct enough, the model is incremented with other interactions, etc., until a satisfying model is obtained. Secondly, the need of *Interaction* instantiation could emerge from the system's dynamics. There we introduce the *Phenomenon* agent which will create or delete interactions in certain conditions. It leads us to consider that an interaction is the manifestation of a phenomenon.

Let us consider two empty compartments (*Entities*) *A* and *B* related by a *DiffusionPhenomenon*. If we add a concentration of a chemical species *C* in compartment *A*, the *Phenomenon* automatically instantiates a *DiffusionInteraction* between *A* and *B* to diffuse *C*. As soon as concentration of *C* is equal in the two compartments, the *DiffusionPhenomenon* destroys the *DiffusionInteraction*.

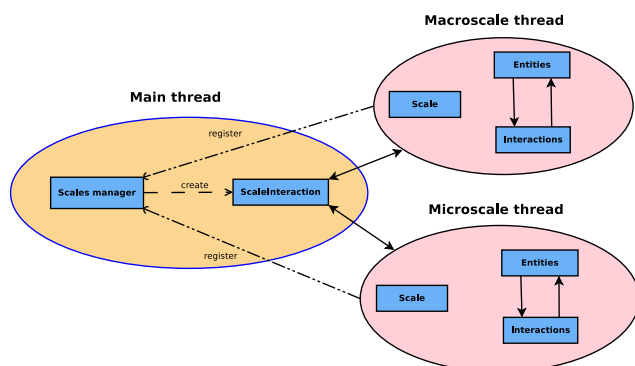


Figure 3: Diagram of threads distribution.

No useless calculation is done this way. Modularity makes this process natural and doesn't require to stop the simulation to modify the code of equations.

Until now, we detailed the different pieces we use to design a single *Scale* using the Virtuo framework. It is time to explain how we proceed to make multiple scales interact into a simulation.

### Multiple interacting scales

Virtuo framework offers the possibility to build simulations composed of several *Scales*. Each one can be run independently in relation to each other. This allows us to improve performances by means of parallelization. As we can see it on figure 3, each *Scale* is distributed into separated threads. They are managed by the *ScalesManager* class that plays the role of a server on which a client *Scale* can log. Once again modularity enables this process to be done while the simulation is running.

We have seen before that multiscale modeling is a useful tool which may be used when a phenomena is not understood or when there is a lack of essential data in a macroscopical model. That is the point that leads us to make multiple *Scales* coexist. These *Scales* must of course be connected so that they can communicate. So we plan to implement this link thanks to a *Scale-Interaction* agent, as well as *Entities* are connected with *Interactions*

This new agent manages exactly two different *Scales*. It is in charge of doing translation work of structure modifications that may appear in each one. There we meet again HMM operators. A *Scale-Interaction* agent provides the two of them : compression from microscale to macroscale and reconstruction from macroscale to microscale. It does itself the translation and acts directly on the two *Scales* components, both *Entities* and *Interactions*.

Such a process requires to ensure data consistency. Indeed *Scales* are not paused and their structures still independantly change in course of the simulation. So we have to use locking mechanisms on *Scales* constituents such as any multi-threaded model. However, they can't guarantee the

system's coherence alone. In fact the autonomy of *Scales* raises some questions about the *Scale-Interaction* action. The following exposes a first and non-exhaustive list of these questions and preliminary answers.

### When should we introduce new microscales ?

As said previously, we need microscales because of a lack of data for solving interactions into the macroscale. This can appear further to a structural evolution induced by the system's dynamics or an intervention from the user. However, microscale simulation implies the choice of a very small time step in relation to spatial units and interactions intensities. Obviously, we can't simulate microscale permanently. So we would only introduce microscales selectively and for a limited time.

### What should we observe to make data estimation ?

Given that a microscale is built from a need to explain explicit parts of a macroscale, the observation is inevitably directed. Thus the microscale's initial state does only contain *Entities* and *Phenomena* we choose to describe it. Though its dynamics could change and drive to take into consideration emerging behaviors then data estimation would mainly be done watching to the *Entities* states and their evolution through the observation duration. But in every instance, it seems that the rules and structure of the microscale must be defined *a priori* and on *ad hoc* basis, in the same way as the macroscale.

### How long should we observe a microscale ?

Since microscales are simulated for a limited time, the observation duration must be short regarding to the macroscale time step. Additionally, the macroscale isn't turned off while the microscale is running. In most cases, the observation would be done until an equilibrium state which can be defined *a priori* or detected as a decrease of interactions intensity.

### How and when to reflect the data estimation ?

Once data estimation is done, *Scale-Interaction* have to reflect it on the macroscale by affecting its structure. This process must be done carefully in order not to provoke instabilities in case of too brutal variations. Sometimes it would be executed progressively and at the right moment which could be difficult to identify.

We have explained principles and problems of the Virtuo framework, from "how to design the model of a scale ?" to "why and how to make scales interacting ?". In the next section, we illustrate our modeling method through a pharmacokinetics / pharmacodynamics application.

## Application to PK / PD of VKA

Our objective on the long term is to provide a virtual laboratory for complex systems in which experts could



build models and conduct experiments. The domain of biochemical kinetics lends itself perfectly to this vision. Moreover, a lot of chemical phenomena involve multiple levels of description. That is why we chose to exemplify the Virtuo framework by means of a PK application.

## Context

The development of a new medicine is very long process which requires clinical trials. They are divided into several stages and lead to identify pharmacokinetics properties of the substance :

- **Absorption** : how it enters the blood circulation
- **Distribution** : how it is disseminated throughout the fluids and tissues of the body
- **Metabolism** : how it is transformed by the body
- **Excretion** : how it is eliminated from the body

Pharmacokinetics may be simply defined as what the body does to the drug, as opposed to pharmacodynamics which may be defined as what the drug does to the body (Benet, 1984). Such practicals aim at measuring adapted dose which should be administered to the patient. They are very expensive and not totally safe. Thus biologists are more and more interested in numerical simulations. The *in virtuo* method allows to be ahead of classical *in silico* simulations thanks to the interactivity with model (Tisseau, 2001). Indeed, multi-agent systems used in the context of a phenomenological approach allows to add sense on phenomena to observe their individual and coupled effects on the system.

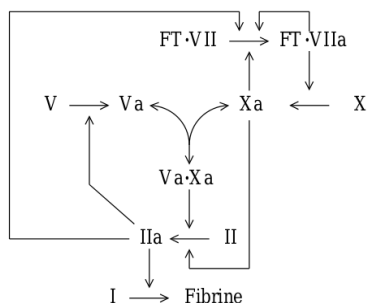


Figure 4: Simplified coagulation cascade (Kerdélo, 2006)

Our study focuses on vitamin K antagonists, a kind of medicine used to cure thrombosis. This work follows those done in the context of *in virtuo* blood coagulation in (Kerdélo, 2006). Blood coagulation, or clotting, is the outcome of a complex reactions cascade that implies coagulation factors ; see figure 4. Generally it arises when a blood vessel is damaged in order to stop the blood loss. Sometimes, this process can be thrown off balance due to a dysfunction of coagulation factors synthesis, which results in clots occurring without any necessity obstructing

the flow of blood through the circulatory system (Abgrall et al., 2004). This synthesis should be regulated in another reactions sequence which can be affected by an excess of vitamin K (figure 5). VKA are prescribed in this instance so as to balance this problem. The chemical reactions entanglements and individual variations of patients make the right dose hard to define (Siguret, 2007). That's why biologists are looking for tools to simulate this process.

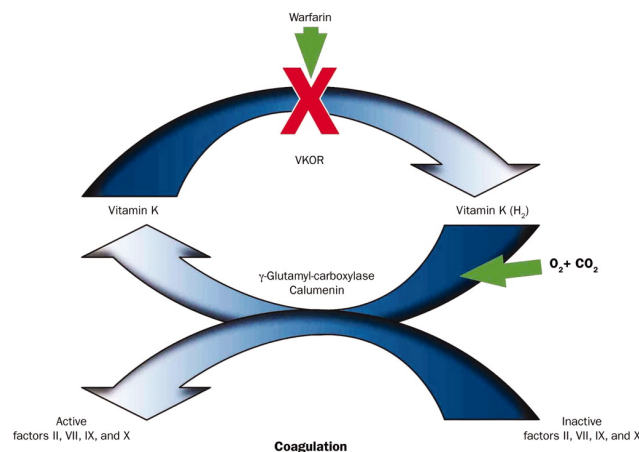


Figure 5: Coagulation factors synthesis (Siguret, 2007)

## Macroscopic model

Classically, PK analysis consists in compartmental models which use kinetics to predict the concentration-time curve in each compartment. More complex PK models, called physiologically-based pharmacokinetics (PBPK) models, rely on the use of physiological information to ease development and validation ; see figure 6. The body is divided into linked compartments that can be associated with black boxes. Inputs and outputs are kinetics parameters which are most often identified by stochastic simulations (Brochot, 2006). It is therefore difficult to understand the various phenomena acting inside this boxes.

We proposed a first MIS implementation of PBPK models in (Le Yaouanq, 2010). We derived the Virtuo framework so as to be able to design chemical systems.

Each compartment is represented by an *Entity*. We linked them by *Diffusion-Phenomena* as in PBPK model. The first novelty of our model comes from the insertion of *Reaction-Phenomena* which operate between concentrations of chemical species inside the compartments. Relations between kinetics and dynamics are evidenced in this way. Our second contribution is based on a realistic identification of parameters of the model. This is where multiscale modeling gets involved. The idea is to simulate redundantly some phenomena in a microscale in order to parameter the macroscale. For the sake of clarity, we only outline our problems on the diffusion rate example.



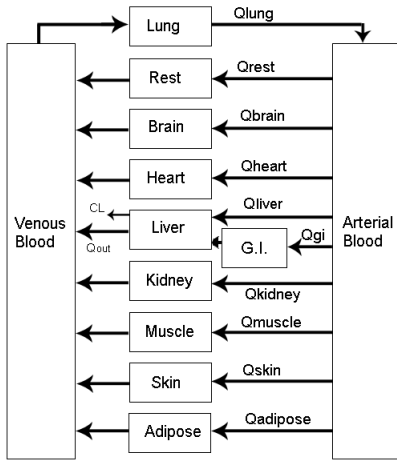


Figure 6: PBPK model (Igari et al., 1983)

**Theoretical elements** Diffusion phenomenon is described by the Fick's laws of diffusion in macroscale (Fick, 1855).

#### Fick's first law

Fick's first law relates the diffusive flux to the concentration, by postulating that the flux goes from regions of high concentration to regions of low concentration, with a magnitude that is proportional to the concentration gradient. In the one dimension case we can write

$$J = -D \frac{\partial C}{\partial x} \quad (4)$$

where :

- $J$  is the diffusion flux,
- $D$  is the diffusion coefficient,
- $C$  is the concentration,
- $x$  is the position.

#### Fick's second law

Fick's second law predicts how diffusion causes the concentration to change with time, on the hypothesis of the matter conservation. Thus the diffused concentration can be computed with :

$$\frac{\partial C}{\partial t}(x, t) = D \frac{\partial^2 C}{\partial x^2}(x, t) \quad (5)$$

This gives rise to the following formula, in the biology perspective, considering two compartments separated by a membrane :

$$\frac{\partial C}{\partial t}(x, t) = \frac{D \cdot S}{L} \cdot \Delta C \quad (6)$$

where :

- $D$  is the diffusion coefficient of a given chemical species at a given temperature,
- $S$  is the surface area over which diffusion is taking place,
- $\Delta C$  is the difference of concentration across the membrane,
- $L$  is the membrane thickness.

**Implementation** A *Diffusion-Interaction*, manifestation of a *Diffusion-Phenomenon*, operates between two compartments. It computes on each time step the diffused concentration from a compartment to the other and applies the modifications.

Let us consider two compartments  $A$  and  $B$  with the diffusion of a chemical species  $C$  from  $A$  to  $B$ . The concentrations of  $C$  in  $A$  and  $B$  from a given instant  $t$  to the instant  $t + 1$  will be altered in this way (using an explicit Euler method for numerical integration)

$$\begin{cases} [C]_A^{t+1} = [C]_A^t - d \cdot \delta t \\ [C]_B^{t+1} = [C]_B^t + d \cdot \delta t \end{cases} \quad (7)$$

where

$$d = \frac{D \cdot S}{L} \cdot ([C]_A^t - [C]_B^t) \quad (8)$$

is the diffused concentration. The figure 7 illustrates this simulation in the context of VKA diffusion.

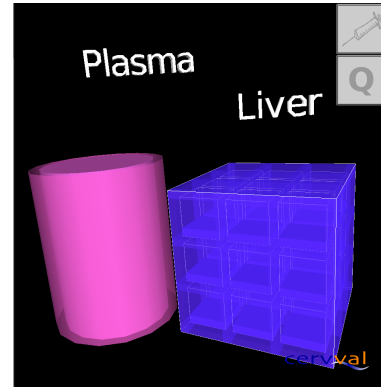


Figure 7: Physiologically-based simulation of VKA diffusion. Colors represent the medicine's concentration and goes from blue (lower) to red (higher).

In this equation, all the parameters can be measured in the simulation except the diffusion coefficient. This piece of data is generally determined with *in vivo* or *in vitro* experimentations for a given temperature and fixed conditions. It is therefore often a missing value in our models. That is why we would like to define this parameter automatically *in virtuo* by the use of a microscale simulation.

## Microscopic model

Diffusion phenomenon at microscale is generally described with the help of stochastic processes (Karatzas and Shreve, 2000). There exist several methods more accurate than the one we expose in the following, but we chose to focus on the principle. We remind in this section some theoretical elements about the Brownian motion and how it is implemented in the Virtuo framework.

**Theoretical elements** Brownian motion, first observed by Robert Brown in 1827, is the random movement of particles suspended in a fluid. It is provoked by collisions of the considered particles with the molecules of fluid which are exposed to thermal agitation.

This random movement leads to a diffusion process which coefficient is given by the Stokes - Einstein law, in case of spherical particles :

$$D = \frac{k_B T}{6\pi\eta R} \quad (9)$$

where :

- $k_B$  is the Boltzmann constant,
- $T$  is the temperature,
- $\eta$  is the fluid viscosity,
- $R$  is the particle radius.

Thus the quadratic displacement of a particle on a  $x$  axis during a time interval  $\Delta t$  is denoted by :

$$\sqrt{\Delta x^2} = \sqrt{2D\Delta t} \quad (10)$$

**Implementation** The *Brownian-Interaction* takes place within a fluid in which particles are immersed. We consider spherical *Entities* which are moveable. The interaction uses then a Gaussian distribution, with a null average and a variance  $\sigma^2 = 2D\Delta t$ , to randomly compute their displacement on each time step (Coulon, 2010).

## Scale interaction

We now have two different levels of description of the diffusion phenomenon. Our aim is to observe the microscale in order to deduce the value of the diffusion coefficient we need in the macroscale. Here's how we proceed.

We introduce a new microscale, based on the state of the macroscale. We arrange randomly particles in a volume according to their concentration in the macroscale. It is the HMM reconstruction operator. We place side by side an empty volume. We add a *Scale-Interaction* agent between the two scales. It counts how many particles crossed from the first to the second compartment, estimates then a diffusion coefficient and sends it to the macroscale ; see figure 8. This is the HMM compression operator. We stop the microscale simulation and continue the macroscale's simulation with the new parameter.

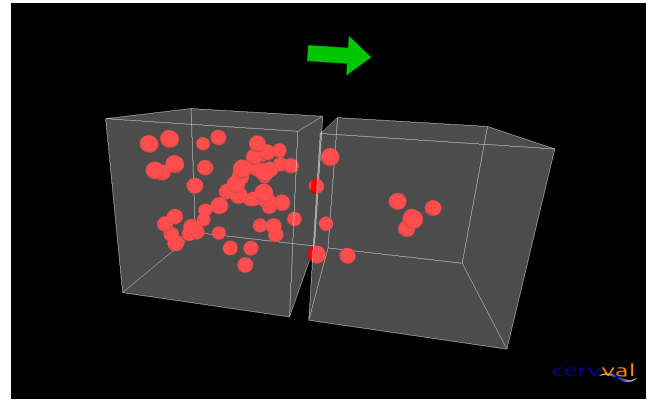


Figure 8: Microscale simulation of diffusion phenomenon using Brownian motion

## Conclusions

We use *in virtuo* experimentation and multi-interactions systems, in the context of complex systems simulation. They enable us to describe phenomena and their actions on the entities composing the system always keeping interactivity with the simulation. This phenomenological approach induces the use of parametric models which parameters are often partially available. This assessment leads us to use multiple levels of description for the phenomena. Thus we simulate redundantly some phenomena at different scales in order to identify the missing parameters.

We propose to implement heterogeneous multiscale methods into the MIS by the introduction of a *Scale-Interaction* agent which plays the role of a translator between the simulated scales.

We illustrate our modeling method through a pharmacokinetics application and a diffusion coefficient identification process. This example points up some remarks and questions we have to answer more precisely in a future work.

Firstly, our will is to parameter a model from observations made on another model. Nevertheless, models aren't perfect by definition. Thus we should keep in mind that what we observe could be imperfect as well.

We can do the same comment about the observation method and data estimation. We inject the macroscale with estimated parameters which could introduce instabilities into the simulation. We need then to define a control mechanism or / and a more developed method to apply observation results.

Secondly, observations are made in the microscale on a very short time window. Indeed, it is generally impossible to run a microscale as fast as a macroscale due to the huge time step difference. But we need results almost immediately to meet the requirements of interactivity with the system. So we are often forced to infer that the observation remains valid for a larger period. It would be satisfactory for

some phenomena but we need to define another observation methodology to be more accurate. For instance, we could try to detect equilibrium states and organizations of *Entities* into the microscale (Ferber et al., 2003). Thus we should be able to partially generalize the observation process even if it seems difficult to define a totally generic method because of the nature of modeling.

## References

- Abgrall, J.-F., Ballet, P., Kerdélo, S., Nicolas, M., Rodin, V., and Tisseau, J. (2004). *L'hématologiste et la coagulation virtuelle*. Margaux Orange.
- Ascher, U. and Petzold, L. (1998). *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. Siam.
- Béal, P.-A., Le Gal, C., and Tisseau, J. (2010). MiMoSA : Mixed Model Simulation Architecture. In *Actes du séminaire du LISyC*.
- Benet, L. Z. (1984). Pharmacokinetics: Basic principles and its use as a tool in drug metabolism. *Drug Metabolism and Drug Toxicity*.
- Brochot, C. (2006). *Paramétrisation de modèles physiologiques toxico / pharmacocinétiques*. PhD thesis, Université Pierre et Marie Curie - Paris 6.
- Coulon, A. (2010). *Stochasticité de l'expression génique et régulation transcriptionnelle. Modélisation de la dynamique spatiale et temporelle des structures multiprotéiques*. PhD thesis, Institut National des Sciences Appliquées de Lyon.
- Debreu, L., Voulard, C., and Blayo, E. (2008). Agrif : Adaptive grid refinement in fortran. *Computers and Geosciences*, 34(1):8 – 13.
- Demazeau, Y. (1995). From interactions to collective behaviour in agent-based systems. In *Proceedings of the First European conference on cognitive science*.
- Desmeulles, G. (2006). *Réifications des interactions pour l'expérimentation in virtuo de systèmes biologiques multi-modèles*. PhD thesis, ENIB.
- Desmeulles, G., Bonneaud, S., Redou, P., Rodin, V., and Tisseau, J. (2009). In virtuo experiments based on the multi-interaction system framework: the réiscop meta-model. *CMES, Computer Modeling in Engineering and Sciences*.
- Ferber, J., Gutknecht, O., and Michel, F. (2003). From agents to organizations: An organizational view of multi-agent systems. In *Agent-Oriented Software Engineering IV*, pages 443–459. Springer Berlin / Heidelberg.
- Fick, A. (1855). Ueber diffusion. *Annalen der physik und chemie*.
- Frenkel, D. and Smit, B. (2001). *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press Inc.
- Fuchs, P., Moreau, G., and Tisseau, J. (2006). *Introduction à la réalité virtuelle*, volume 3 of *Le Traité de la Réalité Virtuelle*, chapter 1, pages 3–32. Presses de l'Ecole des Mines de Paris, 3<sup>e</sup> édition.
- Guibert, D. (2009). *Analyse de méthodes de résolution parallèles d'EDO/EDA raides*. PhD thesis, Université Claude Bernard - Lyon 1.
- Horstemeyer, M. (2009). *Multiscale Modeling : A review*. Springer Science + Business Media.
- Igari, Y., Sugiyama, Y., Sawada, Y., Iga, T., and Hanano, M. (1983). Prediction of diazepam disposition in the rat and man by a physiologically based pharmacokinetic model. *J Pharmacokinet Biopharm*.
- Karatzas, I. and Shreve, S. E. (2000). *Brownian motion and stochastic calculus*. Springer.
- Kerdélo, S. (2006). *Méthodes informatiques pour l'expérimentation in virtuo de la cinétique biochimique. Application à la coagulation du sang*. PhD thesis, ENIB.
- Le Bris, C. (2005). *Systèmes multi-échelles. Modélisation et simulation*. Springer.
- Le Yaouanq, S. (2010). Simulations multi-agents de phénomènes additifs. Application à la pharmacocinétique virtuelle. Master's thesis, École Nationale d'Ingénieurs de Brest.
- Lesne, A. (2003). *Approches multi-échelles pour la physique et la biologie*. Habilitation Diriger des Recherches, Université Pierre et Marie Curie.
- Parenthoën, M. (2004). *Animation phénoménologique de la mer : une approche énaïve*. PhD thesis, Université de Bretagne Occidentale.
- Parisey, N. (2007). *Modélisation et Simulation Multi-Agents de Phénomènes d'Oxydo-Réduction : Application au Complexe III de la Chaîne Respiratoire*. PhD thesis, Université Victor Segalen Bordeaux 2.
- Redou, P., Desmeulles, G., Abgrall, J.-f., Rodin, V., and Tisseau, J. (2007). Formal validation of asynchronous interaction-agents algorithms for reaction-diffusion problems. In *PADS'07, 21st International Workshop on Principles of Advanced and Distributed Simulation*.
- Redou, P., Gaubert, L., Desmeulles, G., Béal, P.-A., Le Gal, C., and Rodin, V. (2010). Absolute stability of chaotic asynchronous multi-interactions schemes for solving ODE. *CMES : Computer Modeling in Engineering and Sciences*.
- Siguret, V. (2007). Antivitamines k et pharmacogénétique : vers une meilleure compréhension de la variabilité individuelle de l'effet dose-réponse. *Pathologie Biologie*.
- Tisseau, J. (2001). *Réalité virtuelle : autonomie in virtuo*. Habilitation à Diriger des Recherches, Université de Rennes 1.
- Weinan, E., Engquist, B., Li, X., Ren, W., and Vanden-Eijnden, E. (2007). Heterogeneous Multiscale Methods: A Review. *Communications in Computational Physics*, 2(3):367–450.

# Encouraging Networks Modularity by Seeding Motifs

Shuguang Li<sup>1</sup>, Jianping Yuan<sup>1</sup> and Juan Cristóbal Zagal<sup>2</sup>

<sup>1</sup>School of Astronautics, Northwestern Polytechnical University, Xi'an, Shaanxi, P.R. China

<sup>2</sup>Department of Mechanical Engineering, University of Chile, Santiago, Chile  
lisg81@gmail.com

## Abstract

We propose a motifs seeding method to encourage the emergence of modular structure during network evolution. Previous studies fail to trigger modularity on freeform evolving ANNs either when varying environmental factors or the evolutionary process itself. We extracted statistical profiles of 3-node and 4-node motifs from evolved networks, and then generated new networks by seeding the most useful 3-node motif (feed-forward loop, ID:38). A series of retina recognition experiments was conducted using the seeded networks. The performance of different algorithms was measured. Our results indicate that modularity could be encouraged under certain conditions. We were able to build networks meeting a desired Z-score.

## Introduction

Modularity is a common property of natural and artificial complex systems. Networked modular structures commonly arise in biology, computer science, social sciences as well as many other disciplines. One can recognize modularity by the presence of clusters of highly interconnected nodes that are sparsely connected to the remaining ensembles of a networked structure (Newman, 2006). Although it is known that modularity is beneficial for the evolvability and robustness of complex systems, its origin remains to be uncovered. The questions of how modularity emerges in complex systems and how it affects the system's performance during development have been frequently addressed (Wagner and Altenberg, 1996).

Artificial evolution provides an excellent platform for exploring the above questions. A variety of systems have been evolved, including simple equational models, expressed by linear matrix transformations, artificial neural networks (ANNs) (Haykin, 1994), representing complex nonlinear phenomena (Yao, 1999), physical simulations involving complex machines and even real robotic systems (Lipson, 2000).

It is generally believed that modularity should be an outcome of an evolutionary process itself. Some experiments have shown that modularity might speed up an evolutionary process (Lipson et al., 2002). Apparently the mechanisms of selection (adequate choice of fitness function), environment variation and noise generation might play a key role in the emergence of modular structures.

Most of previous models used in artificial evolution are relatively simple. Linear models have been used to simplify

the simulations; the nonlinear ANNs models have also been constrained by predefined structures or given building rules. Such limitations could significantly decrease the space of evolutionary search, and simulation results consequently lack of generality.

Freeform ANNs have been employed to increase the generality of systems, but modular networks have not been found under similar experimental conditions at all. Fortunately, further experiments show that the modularity has no conflicts with the evolution of these complex networks (Li and Yuan, 2011). Therefore more effective and general methods have to be designed to encourage the emergence of modularity.

Network motifs are small-scale sub-networks which frequently appear in complex networks, and they have been found in many systems. A network motif can be understood as a pattern or unit of a particular information-processing task. It has been suggested that in many systems the motifs and modularity emerge spontaneously and simultaneously during evolution (Kashtan and Alon, 2005).

Based on these results, we are interested in making use of the coupled mechanism between motifs and modularity, more specifically, in this study we attempt to trigger the appearance of modularity by seeding motifs into ANNs. At first, the motifs' characteristics are extracted from well evolved modular ANNs, and then a series of algorithms is proposed to construct networks with those characteristics. In addition, the well studied retina recognition experiment is conducted in order to make comparisons with previous work.

## Background and Previous Work

A common approach to investigate modularity and its effects on complex systems is to use a computer based simulation of an adaptive system. A model represents the system-environment interplay and a fitness function governs species survival. This computer based method can be seen as a simulation of natural evolution.

Lipson et al. presented a linear matrix abstraction of an adaptive system (Lipson et al., 2002). The linear system represents the transformation of resources and functional requirements for the survival of certain life-form. By randomly varying the elements of a matrix representing the environment, it was possible to observe an increase in the system's modularity. The relation between varying rate and modularity has also been studied by experiments. The authors

claimed that modularity arises in evolutionary systems in response to variation.

Following previous suggestions, research on environment variation was further pursued in (Kashtan and Alon, 2005). A simple feed-forward ANN was used to perform the retina pattern classification task, which had limited connections and a small range of weights. The general structural constraints for evolving the networks were also given. The results show that modularity and motifs spontaneously evolved in networks when the goals were switched in a modular manner during evolution. Their later work also suggested that varying environment could speed up the evolution under certain conditions (Kashtan et al., 2007).

To validate whether HyperNEAT could evolve modular neural networks, Clune et al. investigated a series of retina recognition experiments (Clune et al., 2010), which were similar to those used in previous studies (Kashtan and Alon, 2005). Their results show that HyperNEAT has the potential to produce modular structures in some simple cases, but unfortunately it was unsuccessful in more complex problems. In order to enable HyperNEAT to foster modular networks Verbancsics and Stanley presented a seeding method toward local connectivity, which successfully encouraged the natural emergence of modular structures accelerating the simulations as well (Verbancsics and Stanley, 2010).

Instead of changing the environment, HØerstad proposed the method of adding noise to the genotype-phenotype (G-P) mapping (HØerstad, 2010). He used the same retina recognition experiments to test the noised based methods. The ANNs and their encoding method were similar to those used by Kashtan and Alon (2005). Based on a large amount of simulation experiments, he gave a statistical result, showing that the novel method could trigger the appearance of modularity and finally speed up evolution, however, the switch-goal method does not show the same abilities, which are totally against the conclusions of previous study (Kashtan and Alon, 2005).

Recently, a freeform ANN model has been proposed to investigate the mechanism of modularity and their responses to the variation of environment and evolutionary process. Varying scenarios have been experimented, the results show that the evolution performance has been improved in most cases, however, the modularity never appeared among those scenarios. Further experiments show that the proposed networks have the potential to produce modular networks but more advanced methods are still needed to encourage the emergence of modularity on complex networks (Li and Yuan, 2011).

## Models, Algorithms and Tools

### The ANN Model and Evolutionary Algorithms

To better understand the geometrical properties of complex networks, such as modularity and motifs, we have presented a pure topological ANN (Li et al., 2010), which has binary connection weights and free form directed connections at the hidden layer (Fig. 1). As the architecture shows (Fig. 1), there are three groups of neurons: input neurons, hidden neurons and output neurons, represented as  $I$ ,  $H$  and  $O$  respectively.

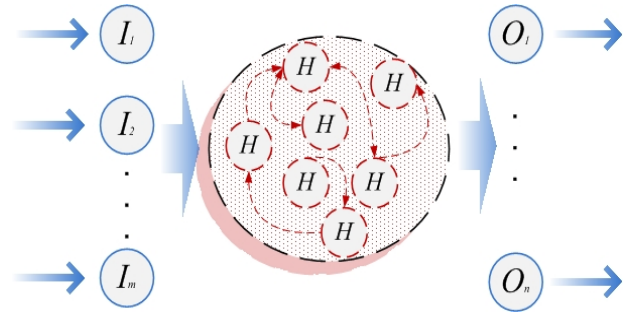


Figure 1: Pure topological neural networks

Therefore, all hidden neurons' and output neurons' values are updated by equation (1) and (2) respectively. The overall model can be given as:

$$H_i(t) = \sin\left(\sum_{j>i} H_j(t-1) + \sum_{f<i} H_f(t) + \sum_r I_r(t)\right) \quad (1)$$

$$O_k(t) = (1 + \exp(-\sum_i H_i(t)))^{-1} \quad (2)$$

where  $H_i$  denotes the current state of the  $i^{\text{th}}$  hidden neuron, which is relative to the other hidden neurons ( $H_j$  and  $H_f$ ) and the input neurons ( $I_r$ ) that connect to  $H_i$ .  $O_k$  is the  $k^{\text{th}}$  neuron's state of all  $n$  output neurons. Due to the characteristics of activation functions, we need to normalize all input raw data into range of  $[0, 2\pi]$  before computation. Accordingly, we have to scale the output value from  $[0, 1]$  to the target range as a final step.

We use the graph encoding method, which directly encodes connections between two nodes in a "from-to" fashion, and then organize all those connections as a graph vector structure. Five evolutionary operators, elitist replication, roulette wheel selection, sub-graph crossover connection mutation and transposing mutation have been used to evolve our networks.

### Modularity Measurement by Artificial Tracer

We measured the modularity of our ANNs using the artificial tracer method (Li and Yuan, 2011), which is inspired by the chemical, isotopic and radioactive tracers. We created the

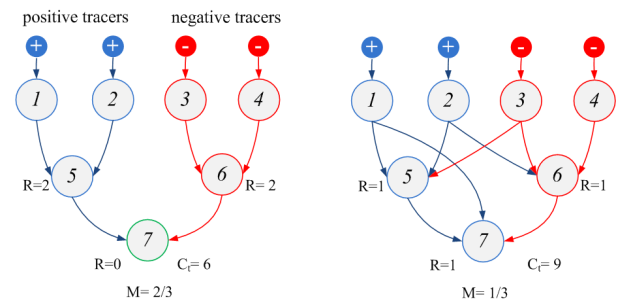


Figure 2: Artificial tracer method

digital tracer elements using different markers, such as positive and negative tracers. To measure the modularity, we first injected the different tracers into each input node of network according to their attributes. All tracers are then passed through other nodes along the directions of information flow. The output connection will pass a tracer to next node with the same marker as the parent node. Annihilation takes place if two tracers with different markers meet at one node. Then we could roughly calculate the modularity using the following equation:

$$M = \frac{\sum_{i=0}^n R_i}{C_i} \quad (3)$$

where  $M$  represents the degree of network modularity, ranging from 0 to 1. Larger values are assigned to networks with higher degree of modularity.  $R_i$  denotes the number of remaining tracers at the  $i^{\text{th}}$  node after annihilations. We summarize their values as the equivalent of total amount of remaining connections. One should notice that the  $R_i$  does not include all input nodes, since the index  $i$  starts counting from the first hidden node.  $C_i$  is the total number of connections within this network. This computation shows the essence of modularity, which is defined as a relation between the inter-connections and intra-connections of elemental modules. An illustration of this procedure is shown in Fig. 2. It should be noted that this method has a limitation for measuring the feedback loop structure.

### The Retina Pattern Recognition Task

We investigated all the scenarios using a classic retina pattern recognition test. The retina pattern recognition experiment has been frequently used in previous studies as a challenging benchmark. Usually, ANNs have been evolved to recognize and classify an artificial retina. Each retina consists of eight pixels (4-pixel wide by 2-pixel height), equally divided into left and right sides, four pixels per side. The goal is to use an ANN to recognize objects in the left and right sides of this retina (Fig.3). As defined in (Kashtan and Alon, 2005), a left object is defined by three or more black pixels or one or two

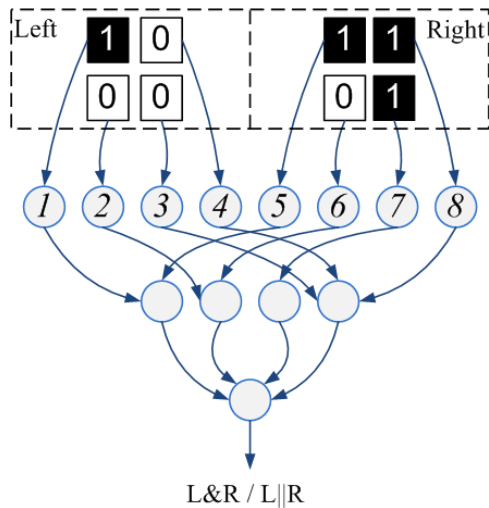


Figure 3: Retina recognition mission

black pixels in the left column only. A right object is defined in a similar way, with one or two black pixels in the right column only. Those eight pixels each could be abstracted as 1 or 0, then those eight binary values could be treated as a group of input signals for the ANN. Finally, the single output (0 or 1) of the ANN is used to decide whether the retina fits the given Boolean logic questions “L AND R”, or “L OR R”. The “L AND R” is true only if the object exists at both sides of the retina, whereas if the object appears in left side or right side or even both sides, the “L OR R” function is then true.

### Motifs Analysis and Seeding

We used the software tools Mfinder (Kashtan et al., 2004) and Fanmod (Wernicke and Rasche, 2006) for extracting the motif’s feature from evolved networks. MDRAW (Kashtan et al., 2004) was used to display the global network topological architecture. As we know (Kashtan and Alon, 2005), a motif’s statistical significance can be described quantitatively using the Z-score.

$$Z_{score} = (N_{real} - N_{rand}) / STD \quad (4)$$

where  $N_{real}$  is the number of times the sub-graph appears in the original network, and  $N_{rand}$  and  $STD$  are the mean and standard deviation of its frequency of appearances in the randomized networks respectively.

---

#### Algorithm 1:

SeedMotifs (Motif\_ID, Target\_Z-score, Net\_size, Max\_refine\_times)

---

```

1: Net_pop ← RandomNetworks(Pop_size)
2: for each Net_i ∈ Net_pop do
3:   App_i ← EnumerateMotifs(Net_i, Motif_ID)
4: end for
5: Mean_app ← Average of App_i in Net_pop
6: STD_app ← Standard deviation of App_i in Net_pop
7: Target_app ← Target_Z-score*STD_app + Mean_app
8: Seeding_model ← Initial
9: Net ← ∅; Current_links ← 0; Current_app ← 0
10: while Current_links < Net_size do
11:   if Seeding_model = Initial then
12:     Net ← MotifSeedInitial (Net, Motif_ID)
13:   else
14:     Net ← MotifSeedRefine (Net, Motif_ID)
15:     Refine_count ← Refine_count + 1
16:   end if
17:   if Current_links ≥ Net_size then
18:     if Refine_count > Max_refine_times then
19:       return Net
20:     else
21:       Current_app ← EnumerateMotifs (Net, Motif_ID)
22:       if Current_app < Target_app then
23:         Reduce_ratio ← 1 - Current_app / Target_app
24:         Net ← ReduceLinks (Net, Reduce_ratio)
25:         Seeding_model ← Refine
26:       else
27:         return Net
28:       end if
29:     end if
30:   end if
31:   Current_links ← LinksCount(Net).
32: end while
33: return Net

```

---



Here we propose a series of algorithms to seed motifs into ANNs and then construct the whole network with expected characteristics. We define the network as  $Net(N, E)$ , where  $N(n_i, i \in [1, Net\_size])$  is the set of all nodes in this network and  $E(e_t, t \in [1, Net\_links])$  represents the set of edges. Each edge  $e_t \{From\_node, To\_node\}$  consists of a connection between two nodes.  $Net\_pop$  is defined as a group of  $Net$ . The  $App$  indicates the appearance time of specific motifs in network. As the algorithm 1 shows, given the motifs' ID and expected Z-score, the function  $SeedMotifs()$  is able to construct a network by repeatedly seeding single type motifs. The feedback of current network's motifs could be obtained by calling the function  $EnumerateMotifs()$ , which will return the appearance time of the motifs, the details of this function are given in algorithm 2. Two types of seeding operators have been designed, which will be used in different stages of seeding. The  $MotifSeedInitial()$  starts at the beginning of the process, whereas, the refining model  $MotifSeedRefine()$  will be executed after reducing the relatively useless links by the  $ReduceLinks()$ . It should be noted that the Algorithm 2 shown here is just for enumerating 3-node motifs, but it can be easily adapted for detecting other motifs. Other major algorithms could be found at the end of this paper.

**Algorithm 2:** EnumerateMotifs( $Net, Motif\_ID$ )

```

1:  $Net\_size \leftarrow$  the size of current  $Net$ 
2:  $En \leftarrow$  the edge number of current  $Motif$ 
3:  $E \leftarrow \emptyset$ 
4:  $Motif\_app \leftarrow 0$ 
5: for each  $Ce \in Net.E$  do
6:    $Ce.degree \leftarrow 0$ 
7: end for
8: for  $i=1$  to  $En$  do
9:    $Em_i \leftarrow false$ 
10: end for
11: for  $a=1$  to  $Net\_size-2$  do
12:   for  $b=a+1$  to  $Net\_size-1$  do
13:     for  $c=b+1$  to  $Net\_size$  do
14:        $E \leftarrow MotifExample(Motif\_ID, n_a, n_b, n_c)$ 
15:       for  $t=1$  to  $En$  do
16:          $Em_t = EdgeMatch(Net, Me_t)$ 
17:       end for
18:       if all  $Em_t = true$  then
19:          $Motif\_app \leftarrow Motif\_app + 1$ 
20:         for each  $Ce \in Net.E$  do
21:            $Ce.degree \leftarrow Ce.degree + 1$ 
22:         end for
23:       end if
24:     end for
25:   end for
26: end for
27: return  $Motif\_app$ 

```

## Experiments and Results

### Experiments on Performances of Algorithms

In order to assess the performance of motifs seeding method, we pursued a group of experiments. We used target networks having 30 nodes and 120 links. We focus our study on seeding 3-node motifs, especially the feed-forward loop motif (ID:38). Given a target Z-score of 10 we have executed 10 independent tests to see the capabilities of seeding speed and convergence, the average Z-score and its standard deviation are shown in Fig.4 (a). It is easy to observe that under limited refining times (10), the Z-score mean rapidly approaches to 10 with a small standard deviation. Furthermore, different Z-score requirements have also been tested, and the 10-times average results are compared with other motif detecting tools as shown in Fig.4 (b). As it can be seen the algorithms perform better on the larger Z-score (5 and 10) targets.

### Experiments on Motifs Extraction

Before seeding the motifs, we have analyzed the modular ANNs by the Fanmod software. All networks were evolved from our previous experiments (Li and Yuan, 2011), resulting in high values of modularity. We extracted all 3-node motifs and some significant 4-node motifs from 10 networks, the statistical results are shown in Fig. 5 and Table.1. From these results, we could observe some simply statistical attributes among all networks. As for 3-node motifs, the motifs with ID of 38 have a mean Z-score of about 20. This means that motif 38 appears significantly more times than others, whereas the motifs 6, 12 and 36 detected from modular ANNs are less than

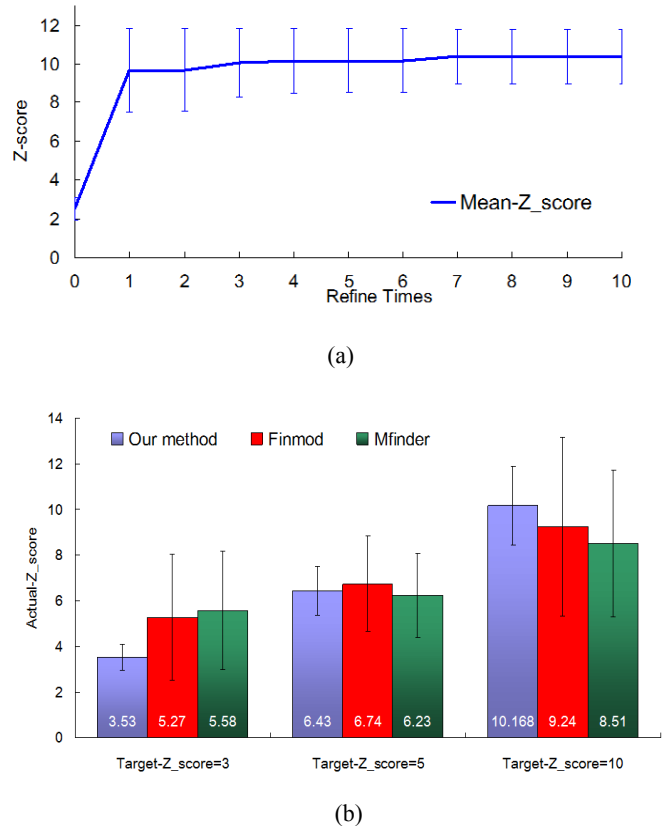


Figure 4: Performances of algorithms: (a) speed and convergence; (b) accuracy



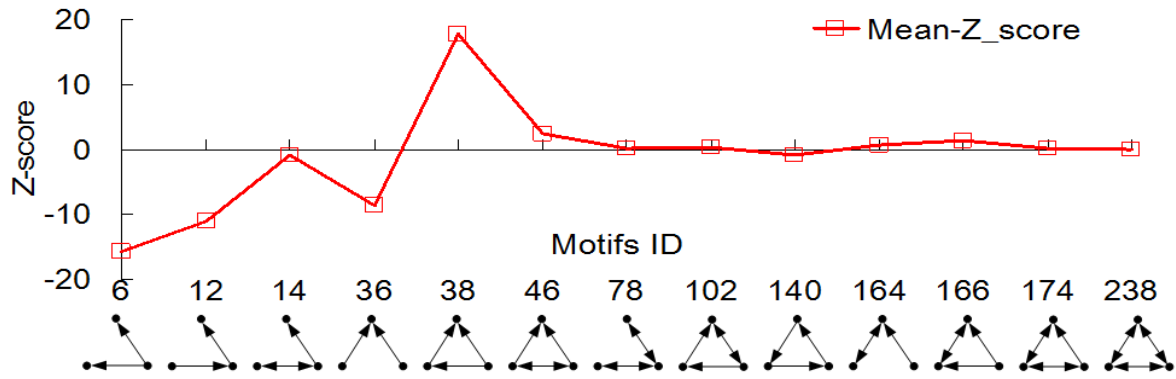


Fig.5. The 3-node motifs' significance profile of networks





	Top two 4-node motifs				Last two 4-node motifs			
Network ID	Motif ID	Z-score	Motif ID	Z-score	Motif ID	Z-score	Motif ID	Z-score
1	2254	9	2252	9.02	2124	-6.73	140	-9.65
2	2254	37.35	2252	20.65	392	-6.71	140	-15.99
3	2254	22.68	2252	18.55	142	-6.79	140	-14.41
4	2254	54.72	2252	40.55	142	-11.89	140	-22.62
5	2254	68.96	2252	46.06	2124	-9.48	140	-24.25
6	2254	6.84	2252	6.18	2124	-4.53	140	-4.88
7	2254	14.26	2252	14.56	2124	-6.92	140	-10.43
8	2254	12.87	2252	11.03	2124	-6.48	140	-12.21
9	2254	7.42	2252	11.13	2124	-5.98	140	-4.362
10	2254	71.7	2252	38.67	2184	-8.49	140	-17.84
Probability	P(2254) = 100%		P(2252) = 100%		P(2124) = 60%		P(140) = 100%	
Average Z-score	 ID:2254 A(2254) = 30.58		 ID:2252 A(2252) = 21.64		 ID:2124 A(2124) = -6.67		 ID:140 A(140) = -13.66	

Table 2: The 4-node Motifs' Significance Profile of Networks

in random networks, we name those 3-node motifs as binary tree motif (ID:6), three-chain motif (ID:12) and reverse binary tree motif (ID:36) respectively. Moreover, the 4-node motifs also show very interesting features. The motifs 2254 (tetrad-feedforward loops motif) and 2252 (bi-feedforward loops motif) appear with highest Z-score among all networks. In contrast, the motif 140 (counter-links four-chain motif) seems to emerge much less than others with a smaller mean Z-score of about -13.66

### Experiments on Retina Recognition

As for the evolutionary simulation, we first constructed a population of 600 candidate networks by seeding the motifs 38 with a target Z-score of 10 and number of links limited to 120. To reduce the computational complexity, we also constrained the network size to 30 nodes from which 8 nodes were assigned as input pixels' values. One node defined as

output, and the remaining nodes (up to 21) were free to build any structures through evolution towards a given task. We set the maximum generation as 5,000 then the modularity was estimated as well as the fitness, and the best networks' structures of each generation were recorded also. In most of our retina recognition experiments, the data set used for training consisted of 100 independent retina patterns which were randomly generated at startup. The general fitness was designed to reflect the ratio of correct recognition over all 100 samples. We evolved the ANNs under a group of different regimes, and we run each test 10 times independently for various experimental scenarios, a list of experiments is shown in Table 1.

We first evolved the networks to recognize the patterns of "L AND R" from the predefined data set. Then, similarly as in previous work, we pursued an interesting MVG regime, in which the recognition goal switched between "L AND R" and

“L OR R” every 50 generations. A varying environment regime (VE) was also tested. We temporally changed the dimension of the data set as a practical method to introduce environment change. Additionally, following the suggestions of (Lipson et al., 2002), we designed the VS scenario as the

Experiment	Description
FG-AND	Evolving networks to solve the fixed goal L AND R
MVG	The goal switched between “L AND R” and “L OR R” every 50 generations.
MVE	The data set changed between 100 samples and randomly selected 50 samples every 50 generations.
VS	The selection mechanism alternated between proportion-based roulette selection and random selection.
VM	The order of mutation operation and selection operation reverse d every 50 generations.
FG-M	Same as FG-AND, but the fitness function coupled with the value of modularity

Table 2: The List of Experiments

variations of selection process, the proportion-based roulette selection mechanism sometimes got a failure during evolution, and then the random selection played a key role for producing offspring. The VM scheme temporally applied the mutational operators after the performance evaluation; it thus reversed the traditional sequence between the selection/replication and the mutation every 50 generations.

The comparisons of results on different regimes are shown in Fig.6, results correspond to the average values over 10 runs. Fig.6 (a) shows the best networks’ fitness records over all regimes, as we can see, the MVE exhibit a significant higher fitness than others, and it approaches 0.95 within 4,000 generations, whereas the MVG does not show any advantages either in fitness value or evolution speed, its fitness value stays under 0.9.

Fig.6 (b) presents the resulting modularity estimation results for the best evolved networks of all regimes. The figure shows a result that a highly modular structure ( $>0.8$ ) which never arose among all previous tests. For most of regimes, the modularity values keep under a low level of 0.5.

Although the mean values of VM cases do not show much advance than others, one of VM tests evolved a highly modular structure and with a high correct ratio about 0.9, the correct ratio and modularity are shown in Fig.6 (c).

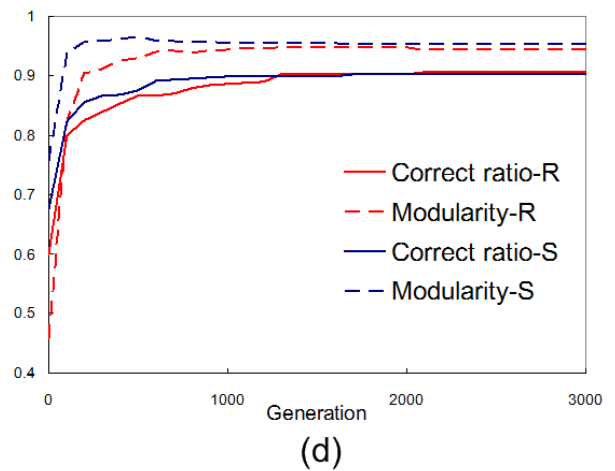
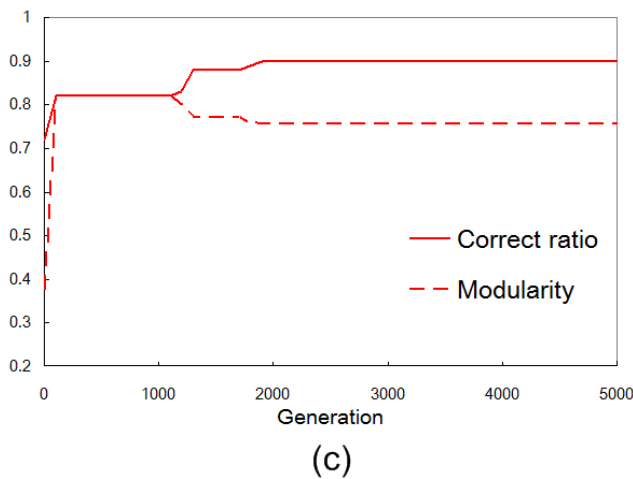
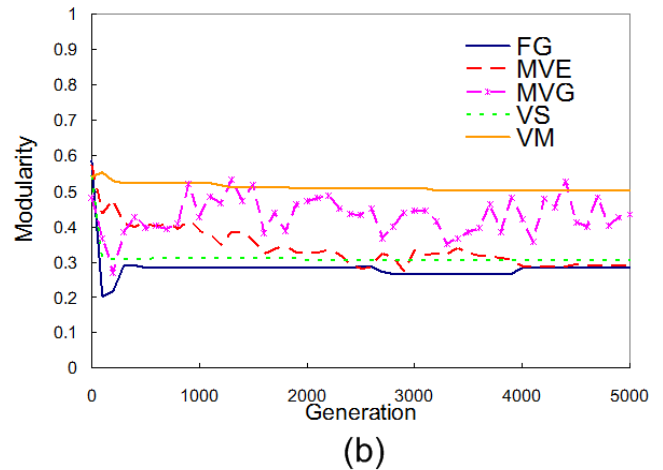
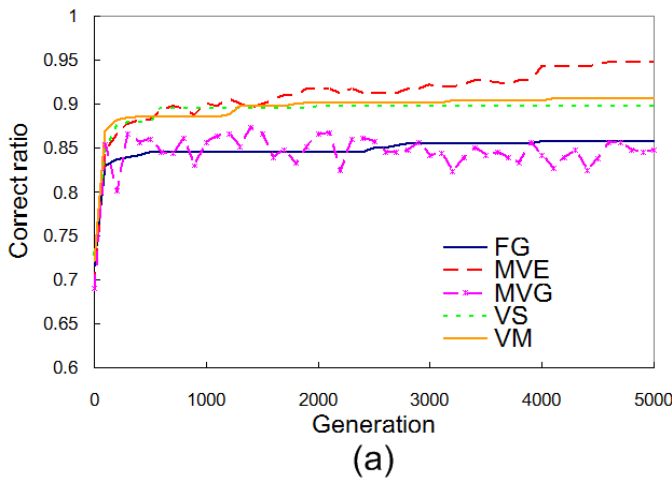


Figure 6: Results of different experiments

As for the FG-M cases, the comparison between random based evolution and motifs-seeding based evolution is shown in Fig.6 (d). It could be easily found that the correction ratio and modularity, they both approach a relative high level at 0.9 and 0.95 respectively. More importantly, these results indicate that the motifs seeding method brings an improvement on the speed of evolution for both correct ratio and modularity.

## Discussion

We have proposed a novel method to construct networks by seeding single type motifs. The performances have been tested by two experiments. Compared to other motifs-detecting tools, our method is able to construct networks with predefined Z-score. The seeding algorithms seem relatively accurate for a higher Z-score ( $\geq 5$ ) and the target Z-score could be quickly achieved within limited iterations. This is mainly attributed to the operators of refining, after reducing the lowest-degree edges, the remaining edges have the opportunities to be reused in new motifs, and then the density of motifs becomes higher. Since we are aiming at seeding a large population, thus the current method is accurate enough, however, we have to admit that the seeding method still have space to be improved on its accuracy, a real-time feedback mechanism might be useful for a more precise seeding.

After analyzing the well evolved modular networks, we have found that for 3-node motifs, the feed-forward loop motif (ID:38) seems very useful for constructing a modular architecture, but the binary tree motif (ID:6), three-chain motif (ID:12) and reverse binary tree motif (ID:36) conflict with modularity. Similar phenomenon was also found for 4-node motifs, where the tetrad-feedforward loops motif (ID:2254) and bi-feedforward loops motif (ID:2252) always appear much more times than others but the counter-links four-chain motif (ID:140) is useless for a modular structure. These results match with previous work very well, it again validates the idea that motifs could emerge spontaneously as the modularity arises, but the hidden mechanism between them still unrevealed. These phenomena are probably due to the natural feed-forward information processing of retina recognition tests, and the inherited relations between full-loop structures (motifs) and their sub-structures (motifs).

According to the analysis of our results, we ran various tests after seeding the feed-forward loop motifs into networks, however in most of cases, the modularity of networks have no improvement compared to our previous work. Fortunately, one of the VM tests has evolved a relatively higher modularity than others. As for all the FG-M cases, the performance of modularity and correct ratio have been both improved, the evolved network (Fig.6(d)) presents a nearly perfect modular structure with a high correct ratio. It is obvious that the emergence speed of modular structures is higher than previous results. These results might be attributed to the motifs seeding mechanism, which offers well organized networks for evolution.

Could the motifs seeding method generate highly modular networks regardless the objective of evolution? Since we just simply seed single motif type into a network, the side-effects of seeding have been ignored, however they might be essential for global performances of networks. Based on this

hypothesis, the multiple-types or hybrid motifs seeding method is needed in the future study.

## Conclusion and Future Work

It is still an open question whether the modularity of ANNs could be encouraged by varying the environment or the evolution process, however, previous work has experimented that the freeform ANNs have difficulty to evolve modular structure under simple variation of external environment.

In this study we try to encourage the networks' modularity by seeding motifs into networks. The motifs statistical features have been extracted from a group of well evolved modular networks. The motif seeding algorithms are proposed and the performances have been evaluated by experiments. We then seeded the network populations by the feed-forward loop motifs and conducted classic retina recognition tests by proposed evolutionary simulation. The modular networks have been discovered during one of tests under varying mutation scenarios. By introducing modularity into fitness function, the modular structures have emerged during evolution; experimental results show that after seeding motifs to initial networks, this emergence process could be accelerated further. These results open the door for triggering modular structure through seeding motifs.

In future, the statistical result will be given based on more experiments under different scenarios. The hybrid motifs seeding algorithms are expected to further encourage the appearance of modularity with a higher success ratio.

## Appendices

---

### Algorithm 4: MotifSeedInitial (*Net*, *Motif\_ID*)

---

```

1:  Net_size  $\leftarrow$  the size of Net
2:  Success  $\leftarrow$  false
3:  while Success = false do
4:    N  $\leftarrow$  Random generate different  $n_a, n_b, n_c$ 
      ( $a, b, c \in [1, \text{Net\_size}]$ )
5:    E  $\leftarrow$   $\emptyset$ 
6:    En  $\leftarrow$  the edge number of current Motif
7:    for  $i=1$  to En do
8:      Emi  $\leftarrow$  false
9:    end for
10:   E  $\leftarrow$  MotifExample(Motif_ID,  $n_a, n_b, n_c$ )
11:   for  $t=1$  to En do
12:     Emt  $\leftarrow$  EdgeMatch(Net, Met)
13:   end for
14:   if all Emi = true then
15:     Success  $\leftarrow$  true
16:     Net.E  $\leftarrow$  E
17:   end if
18: end while
19: return Net

```

---

---

**Algorithm 3:** ReduceLinks (*Net*, *Reduce\_ratio*)

---

```
1: Current_links ← LinksCount(Net)
2: Net.E ← ranked edges by their degrees as a descending order
3: Reduce_start ← Current_links * (1 - Reduce_ratio)
4: for i = Reduce_start to Current_links do
5:   Cei ←  $\emptyset$  (Cei ∈ Net.E)
6: end for
7: return Net
```

---

---

**Algorithm 5:** MotifSeedRefine (*Net*, *Motif\_ID*)

---

```
1: Net_size ← the size of Net
2: Success ← false
3: Total_degree ← degree sum of all edges Ce ∈ Net.E
4: for each Ce ∈ Net.E do
5:   Ce.s_ratio ← Ce.degree / Total_degree
6: end for
7: while Success = false do
8:   Se ← the edge Ce selected by
      roulette mechanism based on s_ratio
9:   na ← Se.from_node; nb ← Se.to_node;
10:  nc ← Random generate nc (c ∈ [1, Net_size], nc ≠ na or nb)
11:  E ←  $\emptyset$ 
12:  En ← the edge number of current Motif
13:  for i = 1 to En do
14:    Emi ← false
15:  end for
16:  E ← MotifExample(Motif_ID, na, nb, nc)
17:  for t = 1 to En do
18:    Emt = EdgeMatch(Net, Met)
19:  end for
20:  if all Emi = true then
21:    Success ← true
22:    Net.E ← E
23:  end if
24: end while
25: return Net
```

---

## References

- Angeline, P. J., Saunders, G. M., and Pollack, J. B. (1993). An evolutionary algorithm that constructs recurrent neural networks. *IEEE Trans. Neural Networks*, 5(1): 54–65.
- Clune, J., Beckmann, B. E., McKinley, P. K., and Ofria, C. (2010). Investigating whether hyperneat produces modular neural networks. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 635–642.
- Floreano, D., Dürr, P., and Mattiussi, C. (2008). Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1(1): 47–62.

- Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ.
- Høverstad, B. A. (2011). Noise and the Evolution of Neural Network Modularity. *Artificial Life*, 17(1):18.
- Kashtan, N., Itzkovitz, S., Milo, R., and Alon, U. (2004). Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics*, 20(11): 1746–1758.
- Kashtan, N., and Alon, U. (2005). Spontaneous Evolution of Modularity and Network Motifs. *Proceeding of National Academy of Sciences*, 102(39):13773–13778.
- Kashtan, N., Noor, E., and Alon, U. (2007). Varying environments can speed up evolution. *Proceeding of National Academy of Sciences*, 104(34): 13711–13716.
- Li, S., Yuan, J., Yue, X., and Luo, J. (2010). The binary-weights neural network for robot control. *Proceedings of the International Conference on Biomedical Robotics and Biomechatronics (BioRob 2010)*, pages 765–770.
- Li, S., and Yuan, J. (2011). The Modularity in Freeform Evolving Neural Networks. *Proceedings of the IEEE Congress on Evolutionary Computation 2011*. pages 2593–2598.
- Lipson, H., Pollack, J.B. (2000). Automated Design and Manufacture of Artificial Lifeforms. *Nature*, 406: 974–978.
- Lipson, H., Pollack, J.B., and Suh, N.P. (2002). On the Origin of Modular Variation. *Evolution*, 56(8):1549–1556.
- Moriarty, D. E. and Miikkulainen, R. (1997). Forming Neural Networks Through Efficient and Adaptive Coevolution. *Evolutionary Computation*, 5(4): 373–399.
- Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceeding of National Academy of Sciences*, 103(23): 8577–8582.
- Verbancsics, P., and Stanley, K. O. (2010). Constraining Connectivity to Encourage Modularity in HyperNEAT, Technical Report, CS-TR-10-10, Department of Electrical Engineering and Computer Science, University of Central Florida.
- Wagner, G. P. and Altenberg, L. (1996). Complex adaptations and the evolution of evolvability. *Evolution*, 50: 967–976.
- Wernicke, S. and Rasche, F. (2006). FANMOD: a tool for fast network motif detection. *Bioinformatics*, 22(9): 1152–1153.
- Yao, X. (1999). Evolving artificial neural networks. *Proceedings of the IEEE*, 87:1423–1447.

# Computational capabilities of small-world Boolean networks

Joseph T. Lizier<sup>1,2,3</sup>, Siddharth Pritam<sup>1</sup> and Mikhail Prokopenko<sup>1</sup>

<sup>1</sup>CSIRO Information and Communications Technology Centre, Locked Bag 17, North Ryde, NSW 1670, Australia

<sup>2</sup>Max Planck Institute for Mathematics in the Sciences, Inselstraße 22, 04103 Leipzig, Germany

<sup>3</sup>School of Information Technologies, The University of Sydney, NSW 2006, Australia  
lizier@mis.mpg.de

## Abstract

We discuss an ensemble investigation of the computational capabilities of small-world networks as compared to ordered and random topologies, using random Boolean functions to provide dynamics of the nodes. We find that the ordered phase of the dynamics (low activity in dynamics) and topologies with low randomness are dominated by information storage, while the chaotic phase (high activity in dynamics) and topologies with high randomness are dominated by information transfer. Information storage and information transfer are somewhat balanced near the small-world regime, providing quantitative evidence that small-world networks do indeed have a propensity to “combine” comparably large information storage and transfer capacity.

## Introduction

It is often suggested that the prevalence of small-world networks in nature is due to an inherent capability to store and transfer information efficiently (Watts and Strogatz, 1998; Latora and Marchiori, 2001). Yet while these claims are all based on quantitative results, they are not based on direct measurement of the relevant dynamic information quantities, either relying on measurements of topological features or on equating perturbation or damage-spreading type results to information transfer. A recently published framework (Lizier et al., 2008, 2010) affords the opportunity to directly measure these computational properties or *information dynamics*.

We discuss our previously published ensemble investigation (Lizier et al., 2011) of the *information dynamics* of small-world Boolean networks, from the perspective of the distributed computation undertaken by the nodes of the network in the transient computation of their attractor. We show that small-world networks exhibit something of a balance between information storage and transfer capabilities, with the capability for apparent (or coherent) information transfer being maximized near the small-world state.

## Small-world Boolean Networks

The small-world network model (Watts and Strogatz, 1998) specifies how to tune networks with  $N$  nodes (with  $\bar{K}$  nearest neighbors each) from ordered, lattice-like structures, to

fully-random topologies using a level of random rewiring of edges  $\gamma$ . There is a significant intermediate range of values  $\gamma$  for which networks exhibit both high clustering (typical of ordered networks) and small average path length (typical of randomized networks); networks in this range are labeled *small-world networks*.

We generate time-series activity for the networks by assigning synchronous random Boolean functions to the nodes (with a bias probability  $r$  for each input configuration of each node to produce a “1” output). This equates to combining *random Boolean networks* (RBNs) (Kauffman, 1993; Gershenson, 2004) with small-world topologies. We select RBNs due to the very large sample space they provide, and their use as models of Gene Regulatory Networks (GRNs). They display a well-known phase transition from *ordered dynamics* (at low connectivity  $\bar{K}$  and activity  $r$ ) to *chaotic dynamics* (at high connectivity and activity), as measured by damage spreading with the normalized Hamming distance  $\delta$  (Gershenson, 2004). We identify the critical state in *finite* networks where the standard deviation  $\sigma_\delta$  of  $\delta$  is maximized. Finally, other recent studies combine RBNs with small-world topologies, e.g. (Lu and Teuscher, 2009).

## Information dynamics

The *active information storage* (Lizier et al., 2010) for a node  $X$  is defined as the average mutual information (MI) between its semi-infinite past  $x_n^{(k)}$  (as  $k \rightarrow \infty$ ) and its next state  $x_{n+1}$  at time step  $n + 1$ :  $A_X(k) = \langle i(x_n^{(k)}; x_{n+1}) \rangle$ .

We note that the local entropy for  $X$  is the sum of  $A_X(k)$  and the local entropy rate  $H_{\mu X}(k) = \langle h(x_{n+1} | x_n^{(k)}) \rangle$ ; i.e.  $H_X = A_X(k) + H_{\mu X}(k)$ . In a deterministic system such as RBNs, there is no intrinsic uncertainty in  $H_{\mu X}(k)$  so it represents the *joint* contribution or transfer from the causal information sources to the destination (Lizier et al., 2010).

The *information transfer* (formulated in the *apparent transfer entropy* (Schreiber, 2000; Lizier et al., 2008)) from *one* source  $Y$  to a destination  $X$  is the average MI between the previous source state  $y_n$  and the next destination state  $x_{n+1}$ , *conditioned* on the semi-infinite past of the destination  $x_n^{(k)}$  (as  $k \rightarrow \infty$ ):  $T_{Y \rightarrow X}(k) = \langle i(y_n; x_{n+1} | x_n^{(k)}) \rangle$ .

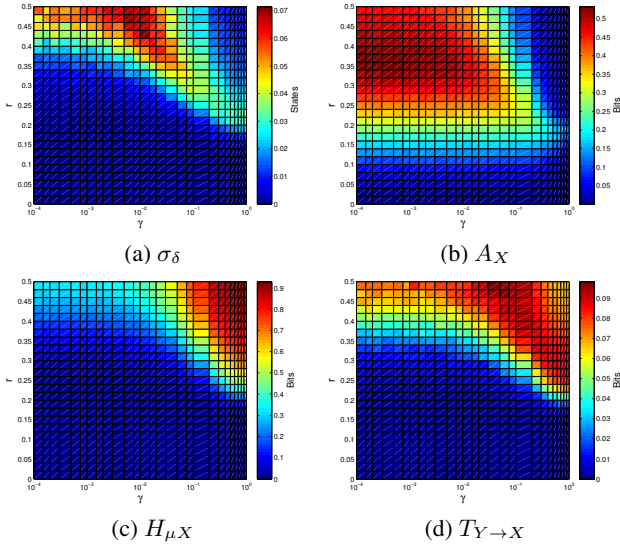


Figure 1: Measures of dynamics versus  $r$  and  $\gamma$  for  $K = 4$  (color online). Ordered dynamics occur for low  $r$  and  $\gamma$  (bottom left), and chaotic dynamics for high  $r$  and  $\gamma$  (top right); the critical regime at the maxima in (a) separates these.

## Results and discussion

We examine ensembles of networks of size  $N = 264$  as a function of  $\bar{K}$ ,  $r$  and  $\gamma$  (sources of links only rewired; other details in (Lizier et al., 2011)) using extensions to RBNLab (Gershenson, 2003). We use  $\bar{K} = 4$ : the small-world region then occurs approximately for  $0.03 \leq \gamma \leq 0.1$ .

Using  $\sigma$  (not shown) and  $\sigma_\delta$  (Fig. 1a, Fig. 2) we see that (i) for fixed  $\gamma$  an ordered phase exists for low  $r$ , with the chaotic phase for large  $r$ . Crucially, a similar transition occurs (ii) with respect to  $\gamma$  for fixed  $r$ , with ordered dynamics for small  $\gamma$  (more ordered networks), and chaotic dynamics for large  $\gamma$  (more randomized networks). The critical region in dynamics has much similarity to the small-world regions of  $\gamma$  (however this is highly dependent on activity  $r$ ).

The ordered phase of these dynamics (low  $r$  and  $\gamma$ ) is dominated by information storage  $A_X$  (Fig. 1b), while the chaotic phase of the dynamics (high  $r$  and  $\gamma$ ) is dominated by information transfer (captured in total in  $H_{\mu X}$  in Fig. 1c). The critical regime exhibited a balance between these two operations (Fig. 2), and since this was near to the small-world topology regime, it could be said that **small-world networks have a propensity to combine comparably large information storage and transfer capabilities**.

This balance can be explained by considering how the topological features related to the information dynamics. Information storage is strongly correlated to the clustering coefficient: locally clustered structure appears to strongly support storage operations. In contrast, information transfer was anti-correlated with average path length: long links appear to be a crucial facilitator of transfer across the network.

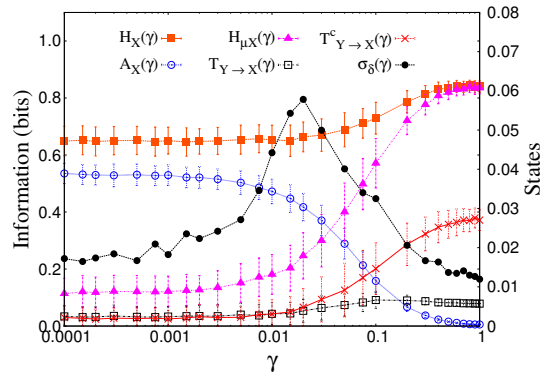


Figure 2: Measures of dynamics versus  $\gamma$ , for  $K = 4$  and  $r = 0.36$ .  $\sigma_\delta$  is plotted against the right y-axis. Error bars indicate *standard deviation* across 250 sampled networks.

Additionally, Fig. 1d shows apparent information transfer  $T_{Y \rightarrow X}$  is maximized slightly inside the chaotic phase of dynamics (near to the small-world regime). The capacity for coherent computation is eroded as too many random links promote interactions and make the dynamics more chaotic.

These results add evidence that small-world networks hold computational advantages over other topologies.

## References

- Gershenson, C. (2003). RBNLab. Online software: <http://rbn.sourceforge.net>.
- Gershenson, C. (2004). Introduction to random Boolean networks. In Bedau, M., Husbands, P., Hutton, T., Kumar, S., and Suzuki, H., editors, *Proceedings of the Workshops and Tutorials of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALife IX)*, Boston, USA, pages 160–173.
- Kauffman, S. A. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, New York.
- Latora, V. and Marchiori, M. (2001). Efficient behavior of small-world networks. *Physical Review Letters*, 87(19):198701.
- Lizier, J. T., Pritam, S., and Prokopenko, M. (2011). Information dynamics in small-world Boolean networks. *Artificial Life*. to appear.
- Lizier, J. T., Prokopenko, M., and Zomaya, A. Y. (2008). Local information transfer as a spatiotemporal filter for complex systems. *Physical Review E*, 77(2):026110.
- Lizier, J. T., Prokopenko, M., and Zomaya, A. Y. (2010). Information modification and particle collisions in distributed computation. *Chaos*, 20(3):037109.
- Lu, Q. and Teuscher, C. (2009). Damage spreading in spatial and small-world random Boolean networks. arXiv:0904.4052.
- Schreiber, T. (2000). Measuring information transfer. *Physical Review Letters*, 85(2):461–464.
- Watts, D. J. and Strogatz, S. (1998). Collective dynamics of ‘small-world’ networks. *Nature*, 393:440–442.

# Controlling Legged Robots with Coupled Artificial Biochemical Networks

Michael A. Lones<sup>1,2</sup>, Andy M. Tyrrell<sup>1,2</sup>, Susan Stepney<sup>1,3</sup> and Leo S. Caves<sup>1,4</sup>

<sup>1</sup>York Centre for Complex Systems Analysis (YCCSA)

<sup>2</sup>Department of Electronics, <sup>3</sup>Department of Computer Science, <sup>4</sup>Department of Biology

University of York, Heslington, York YO10 5DD, UK

{mal503,amt}@ohm.york.ac.uk, susan.stepney@cs.york.ac.uk, leo.caves@york.ac.uk

## Abstract

Artificial biochemical networks (ABNs) are computational architectures motivated by the organisation of cells and tissues at a biochemical level. In previous work, we have shown how artificial biochemical networks can be used to control trajectories in discrete and continuous dynamical systems. In this work, we extend the approach to the control of a hybrid dynamical system: a legged robot. Taking inspiration from biological cells, in which complex behaviours come about through the interaction of different classes of biochemical network, we develop the notion of a coupled artificial biochemical network, in which an artificial genetic network controls the configuration of an artificial metabolic network. Using a higher-level robotic control task, we show how the coupled network finds solutions which can not be readily expressed using the artificial genetic network or artificial metabolic network alone. Our results also show the important role that non-linear maps can play as a natural source of complex dynamics.

## Introduction

The structure and function of biological organisms emerges from the action and interaction of biochemical networks operating within cells. There are three main types of biochemical network: the *metabolic network*, comprising the protein-mediated chemical reactions that take place within the cell; the *signalling network*, composed of the protein-mediated responses to chemical messengers received by the cell; and the *genetic network*, which emerges from the regulatory interactions between genes.

From a computational perspective, biochemical networks are interesting for a number of reasons. This includes their ability to express complex behaviours, their compactness, their ability to adapt to changing environments, their robustness to environmental perturbation and—from the perspective of evolutionary computation—their evolvability. Such reasoning has motivated a host of computational models whose architectures are based upon the structure and function of biochemical networks. We refer to these collectively as *artificial biochemical networks*, or ABNs (Lones et al., 2010).

Perhaps best known of these are Boolean networks (Kauffman, 1969) and other kinds of *artificial genetic networks* (e.g. Reil, 1999; Banzhaf, 2003). By modelling the regulatory interactions which occur between genes, these models attempt to capture the dynamics of genetic networks, using these to generate complex, robust, behaviour. Another class of models, which includes P Systems (Păun, 2000) and artificial chemistries (e.g. Fontana, 1992; Banzhaf, 2004), can be categorised as *artificial metabolic networks*. These mimic the self-organising behaviour of biological metabolisms, and attempt to capture the manner in which complex behaviour can emerge from interactions between simple computational components. There has also been some work on *artificial signalling networks*, including early work on perceptron-like feed-forward networks (Bray, 1995) and more recent work on signalling-based classifier systems (Decraene et al., 2007).

ABNs have been used to implement a range of computational behaviours, including those required for robotic navigation (Ziegler and Banzhaf, 2001; Taylor, 2004), classification (Banzhaf and Lasarczyk, 2005), pole balancing (Nicolau et al., 2010) and image compression (Trefzer et al., 2010). In our research, we are interested in the ability of ABNs to control the kind of dynamics found in complex real world systems. In (Lones et al., 2010), we applied ABNs to the control of two numerical dynamical systems: the Lorenz equations, a continuous-time dissipative dynamical system; and Chirikov's standard map, a discrete-time conservative dynamical system. These both model complex dynamics found within real world systems, and also lie at opposite ends of the dynamical systems spectrum. In both cases, we were able to evolve ABNs capable of controlling trajectories in a prescribed manner.

However, many real world systems do not have purely continuous or discrete dynamics, but rather a hybrid of the two (Branicky, 2005). These often occur on different time scales, such that continuous state flow is occasionally interrupted by jump discontinuities caused by the occurrence of discrete events. Two common examples of this are physical systems with impact, such as a bouncing ball, and switched



systems, where a signal change causes a discrete change in behaviour. In this paper, we consider a problem which combines both of these: controlling the gait and direction of movement of a simulated legged robot.

Coupling between different classes of biochemical network plays an important part in the functioning of biological cells. The coupling between a genetic network and a metabolic network, in particular, is central to a cell's ability to both specialise and adapt to a changing environment. Taking inspiration from this biological behaviour, we investigate a hybrid ABN architecture, in which an artificial genetic network controls the expression of an artificial metabolic network. Results on the robot locomotion tasks suggest that such an architecture is particularly suited to problems that require reconfigurable dynamical behaviour.

The paper is structured as follows: We first introduce the ABN models used in this work. We then describe how these models are evolved. Finally, we introduce the robotic locomotion tasks to which they were applied, and present results and conclusions.

## Artificial Biochemical Network Models

In this section, we describe the three ABN models used in this work: an artificial genetic network (AGN), an artificial metabolic network (AMN), and a hybrid ABN formed from the coupling of an AGN and an AMN. In addition to expressiveness and evolvability, our choice of models is also influenced by a desire for efficiency and simplicity. For this reason, the models use discrete-time rather than continuous-time updates (unlike, for instance, Banzhaf, 2003). Since continuous-time dynamical systems can often be reduced to discrete-time equivalents by taking Poincaré sections (Kantz and Schreiber, 2004), this arguably makes little difference in terms of expressiveness, but does considerably reduce execution time.

### Artificial Genetic Network (AGN)

In general, the complex behaviour of biological genetic networks stems not from the complexity of their component parts, but from the complexity of their dynamics. Hence, a simple abstraction such as the Boolean network can display complex behaviour without the need to model biological details such as continuous-valued expression, asynchronous updates, continuous-time, and the presence of transcription factors. Nevertheless, there are advantages to using more complicated models, and in this work we use a continuous-valued generalisation of the Boolean network.

Continuous values have two main advantages. First, they make it easier to interface with external systems, since inputs and outputs do not need to be encoded in binary. Second, the size of the state space is not limited by the number of genes in the network. In a Boolean network, the number of possible states is  $2^N$ , where  $N$  is the number of genes, meaning that small networks are always attracted to a limit

cycle. When continuous values are used, the state space is infinite (within the limits of representation), meaning that small networks have the potential to exhibit more complex behaviours.

Formally, an AGN consists of an indexed set of genes,  $G$ . Each  $g_i \in G$  has an expression level  $\lambda_i$ , an indexed set of regulatory inputs  $R_i$ , and a regulatory function  $f_i$ , which maps the expression levels of its regulatory inputs to its own expression level. The first time the AGN is executed, its expression levels are initialised from an indexed set of initial values,  $L_G$ . External inputs can be delivered to the network either by explicitly setting the expression levels of certain genes, or by introducing new regulatory inputs with fixed values. After iterating the network a specified number of times,  $t_G$ , outputs are captured from the final expression levels of designated genes.

### Artificial Metabolic Network (AMN)

The artificial metabolic network complements the AGN described in the previous section. It is a simple artificial chemistry with continuous-valued chemicals and continuous-valued reactions. Formally, it consists of an indexed set of enzyme-analogous elements  $E$  which transform the concentrations of an indexed set of real-valued chemicals  $C$ . Each enzyme has a set of substrates  $S_i \subseteq C$ , a set of products  $P_i \subseteq C$ , and a reaction  $m_i$  which calculates the concentrations of its products based upon the concentrations of its substrates.

The first time the AMN is executed, its chemical concentrations are initialised from an indexed set of initial values,  $L_C$ . External inputs are delivered to the network by explicitly setting the concentrations of certain chemicals. At each time step, each enzyme  $e_i$  applies its reaction  $m_i$  to the current concentrations of its substrates  $S_i$  in order to determine the new concentrations of its products  $P_i$ . Where the same chemical is produced by multiple enzymes, i.e. when  $\exists j, k : j \neq k \wedge c_i \in P_j \cap P_k$ , the new concentration is the mean output value of all contributing enzymes:  $c_i = \sum_{e_j \in E_{c_i}} c_i^{e_j} / |E_{c_i}|$  where  $E_{c_i}$  are enzymes for which  $c_i \in P_i$  and  $c_i^{e_j}$  is the output value of  $e_j$  for  $c_i$ . After iterating the network  $t_M$  times, outputs are captured from the final concentrations of designated chemicals.

### Coupled Artificial Biochemical Network (CABN)

Biological biochemical networks interact with one another in a number of ways. Perhaps most significantly, the genetic network controls when and where proteins are expressed. This determines which enzymes are present in the metabolic network, and hence which reactions can take place within a cell. In effect, the genetic network is able to reconfigure the cell's processing machinery over the course of time. This behaviour occurs extensively in both single-celled and multicellular organisms. In the former, it allows the metabolism to be changed in order to react to the presence of different

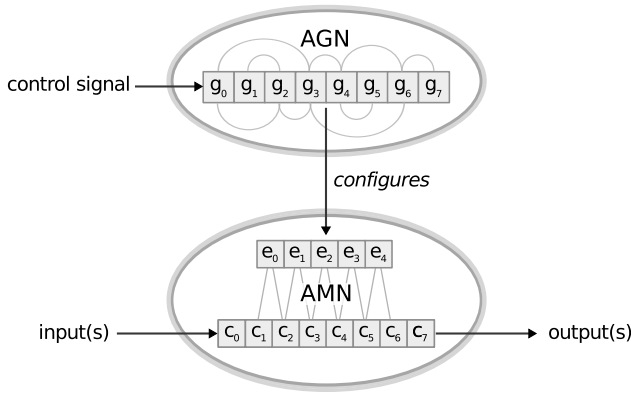


Figure 1: Coupled artificial biochemical network.

kinds of nutrients in the organism’s environment. In the latter, it underlies the processes of cell specialisation and development which are fundamental to multi-cellular organisms.

In the coupled artificial biochemical network (CABN) model, we capture this idea of a genetic network controlling the expression of a metabolic network (See Fig. 1). Formally, a CABN comprises an AGN, an AMN, and an injective coupling function  $\chi : G_C \rightarrow E$  where  $G_C \subseteq G$  is the set of enzyme coding genes, i.e. each enzyme is coupled to a single gene, and some genes may not be enzyme coding (yet are still involved in regulating other genes). Coupling is carried out by giving each enzyme an expression level,  $\xi_i$ , and setting this to the expression level of the gene to which it is coupled, i.e.  $\forall (g_i, e_j) \in \chi : \xi_j := \lambda_i$ . This expression level then determines the relative influence of each enzyme when calculating the new concentration of a chemical:

$$c_i = \sum_{e_j \in E_{c_i}} \frac{\xi_i c_i^{e_j}}{\sum_{e_j \in E_{c_i}} \xi_i} \quad (1)$$

i.e. the new concentration is the mean of each enzyme’s output value weighted by its relative expression level. This captures the idea that changes in the genetic network lead to changes in the balance between competing pathways in a metabolism.

## Regulatory functions and enzyme reactions

Table 1 lists the mathematical functions from which regulatory functions ( $f$ ) and enzyme reactions ( $m$ ) are chosen.

*Sigmoids* model the switching behaviour of many non-linear biological systems, making them a good choice for approximating the behaviours of genetic and metabolic pathways. We use the logistic function, where  $s$  determines the slope and  $b$  the slope offset (or *bias*). For multiple inputs,  $x = \sum_{j=0}^n i_j w_j$ , where  $i_0 \dots i_n$  are inputs and  $w_0 \dots w_n \in [-1, 1]$  are corresponding input weights, with negative values indicating repression.

The remaining functions, all of which are discrete non-linear maps, are motivated by our earlier work (Lones et al.,

Table 1: Mathematical functions used within ABNs.

---

Logistic (Sigmoidal) function:

$$f(x) = (1 + e^{-sx-b})^{-1}, \text{ where } s \in [0, 20], b \in [-1, 1]$$

Logistic map:

$$x_{n+1} = rx(1-x), \text{ where } r \in [0, 4]$$

Arnold’s cat map:

$$(x_{n+1}, y_{n+1}) = ([2x_n + y_n] \bmod 1, [x_n + y_n] \bmod 1)$$

Baker’s map:

$$(x_{n+1}, y_{n+1}) = \begin{cases} (2x_n, y_n/2) & 0 \leq x_n \leq \frac{1}{2} \\ (2 - 2x_n, 1 - y_n/2) & \frac{1}{2} \leq x_n < 1 \end{cases}$$

Chirikov’s standard map:

$$\begin{aligned} p_{n+1} &= (p_n + K \sin \theta_n) \bmod 2\pi, \quad K \in [0, 10] \\ \theta_{n+1} &= (\theta_n + p_{n+1}) \bmod 2\pi \end{aligned}$$


---

2010) in which we found that the use of logistic maps within ABNs could lead to the evolution of more effective controllers. We hypothesised that this was due to evolution taking advantage of the complex dynamical behaviours displayed by non-linear discrete maps.

In this work, we extend the approach by using four well-known discrete maps that capture the natural dynamics present in a range of biological and physical systems. *The logistic map* is a model of biological population growth. Depending on the value of parameter  $r$ , the system is attracted to either a fixed-point, cyclic or chaotic orbit (May, 1976). *Arnold’s cat map* (Arnold and Avez, 1968) is a geometric transformation of the unit square with interesting periodic behaviour. *The baker’s map* is an archetypal model of deterministic chaos, capturing the exponential sensitivity to initial conditions that results when kneading bread (Silva, 2008). *Chirikov’s standard map* (Chirikov, 1969) captures the behaviour of dynamical systems with co-existing ordered and chaotic regimes. Its dynamics are ordered for low values of parameter  $K$  and become increasingly chaotic for higher values. The parameterised maps (the logistic map and Chirikov’s map) can be used either with an evolved parameter value or with an extra input, whose current value is used to set the parameter. The latter is referred to as a tunable map, since its dynamics can be modified by the ABN during execution.

## Evolving Artificial Biochemical Networks

Our ABNs are evolved using a standard generational evolutionary algorithm with tournament selection (size 4), uniform crossover ( $p=0.15$ ), and point mutation ( $p=0.06$ ). Crossover points always fall between gene or enzyme boundaries. Inputs and outputs ( $R_i, S_i$  and  $P_i$ ) are represented by absolute references to indices. Function parameters (e.g. slopes, input weights) and initial values are rep-

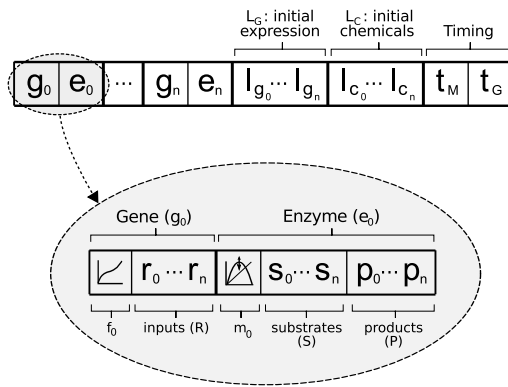


Figure 2: Genetic encoding of an artificial biochemical network.

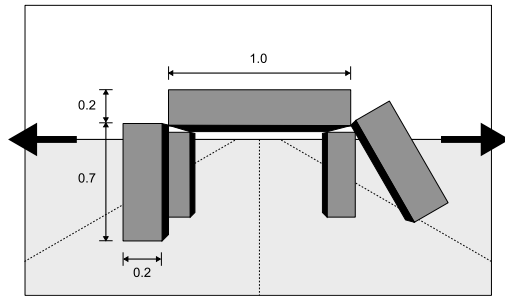


Figure 3: Quadruped robot simulated in Open Dynamics Environment. Arrows indicate the direction of movement along the x-axis plane.

resented as floating-point values and are mutated using a Gaussian distribution centred around the current value.

We use a standardised genetic encoding for all ABN types (see Fig. 2). This represents the ABN as a sequence of genetic units, where each genetic unit has an optional regulatory region and an optional coding region. In a coupled network, the regulatory region encodes the gene and the coding region encodes the enzyme which it expresses. Where a gene does not express an enzyme (such as in an AGN), the coding region is empty. For an AMN, where there are no genes, the regulatory region is empty. The genetic encoding also includes the initial gene expression and chemical concentrations (where applicable) and timing information.

## Controlling Legged Robot Locomotion

Legged robot locomotion is a challenging problem. In (Beer and Gallagher, 1992), the authors summarised the challenge by stating “A locomotion system must simultaneously solve the two tightly coupled problems of support and progression.” In this paper, we address the locomotion of a simulated quadrupedal robot. There have been a number of previous attempts to evolve quadrupedal locomotion (e.g. Hornby et al., 2005; Kamio et al., 2003; Seo and Hyun, 2008; Clune et al., 2009). Since functional gaits can be generated by tapping sinusoidal functions at appropriate phase offsets,

a common approach is to use genetic algorithms (Hornby et al., 2005) or genetic programming (Seo and Hyun, 2008) to generate sinusoid-based controllers. Another, potentially more robust, approach is to evolve neural networks (Beer and Gallagher, 1992; Clune et al., 2009).

Since our focus is upon using legged robot locomotion as a test bed for comparing the expressiveness of different ABN models, the robot (see Fig. 3) is purposely very simple in design, comprising a square top section with four legs connected by actuators at the corners. The actuators are limited to movement in the x-axis plane, with a maximum elevation of  $60^\circ$  from vertical. The robot was simulated using the Open Dynamics Engine (ODE) physics engine, with a step size of  $\Delta_t = 0.05s$ , friction of 200N, CFM (an ODE parameter) of  $10^{-5}$ , and standard gravity. Actuators have a maximum angular velocity of 3m/s and a maximum torque of 150Nm. These values are sufficient to enable dynamic gaits, but not large enough to allow the body to be dragged by the front legs without the involvement of the rear legs. The ABN is executed every 10 simulation steps.

## Generating Quadrupedal Gaits

The first task was to evolve ABNs capable of generating quadrupedal gaits, i.e. patterns of actuator movements which would cause the robot to move away from its starting position. The aim of this task was to determine whether the different ABN types and configurations were able to generate appropriate patterns of movement.

**Experimental Settings** A controller’s fitness is the Euclidean distance moved by the robot within an evaluation period of 500 time steps. The population size is 200, with a generation limit of 100. ABNs have four inputs, corresponding to the actuator angles, and four outputs, which are used to set the torques of the actuators during the next 10 simulation steps. Note that the requirement to map angles to torques adds a degree of difficulty to this task. All inputs and outputs are linearly scaled to the interval  $[0, 1]$ . For AMNs and CABNs, inputs are delivered via initial chemical concentrations. For AGNs, inputs are delivered via initial gene expression levels.

**Results** Figure 4 compares the fitness distributions of evolved controllers. This shows that all three classes of ABN are capable of generating gaits which solve the movement task. It also indicates that there is no significant difference in the median performance of the AGN, AMN and CABN models. However, for all ABN models, the best controllers use Sigmoidal functions rather than non-linear maps. Solution length (i.e. network size) has relatively little impact. Examples of evolved behaviours are shown in Figure 5.

These results demonstrate that effective controllers can be expressed using any of the ABN models, although good controllers are more readily found when using Sigmoidal functions. It is interesting to note that there is no observable

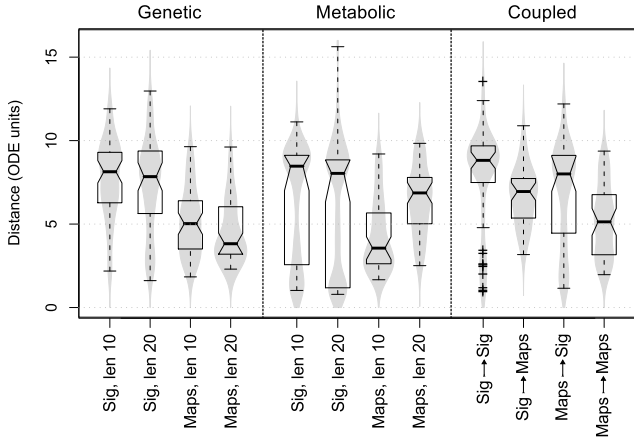


Figure 4: Controlling legged robots using coupled and uncoupled ABNs with sigmoids (Sig) or discrete maps. Summary statistics for 50 runs are shown as notched box plots. Overlapping notches indicate that median values (thick horizontal bars) are not significantly different at the 95% confidence level. Kernel density estimates of underlying distributions are also shown (in grey), showing that some of the distributions are multimodal. The notation  $F_{n1} \mapsto F_{n2}$  denotes a genetic network with  $F_{n1}$  regulatory functions coupled to a metabolic network with  $F_{n2}$  enzyme functions. Coupled networks comprise 10 genes (expressing up to 10 enzymes) and 10 chemicals. For uncoupled genetic and metabolic networks, results are shown for solution lengths of both 10 and 20 (genes, or enzymes and chemicals, respectively), to allow fair comparison with the coupled networks.

penalty to using the structurally more complex coupled networks.

### Higher Level Control of Locomotion

The second task introduced an extra level of difficulty, requiring the ABNs to control the robot's direction of movement in addition to its gait. The aim of the task was to test not only the ABNs' abilities to express suitable patterns of movement, but also their ability to switch between different patterns as required.

**Objective function** The robot is required to change direction by  $180^\circ$  when signalled to do so, whilst still moving as far as possible in the given direction. Controller fitness is measured over a sequence of epochs  $\langle e_0, \dots, e_{N-1} \rangle$ , each with a random duration between 300 and 600 time steps, with the required direction of movement reversing during subsequent epochs. The fitness function  $f$  is defined:

$$f = \frac{t_{\max} - t_{\min}}{N} \min \left\{ \sum_{n \in \mathbb{N}_{\text{even}}, n < N} p(n), \sum_{n \in \mathbb{N}_{\text{odd}}, n < N} p(n) \right\} \quad (2)$$

where  $t_{\max}$  and  $t_{\min}$  are the maximum and minimum bounds on epoch duration and  $p(n)$  is the progress made during epoch  $n$ , defined:

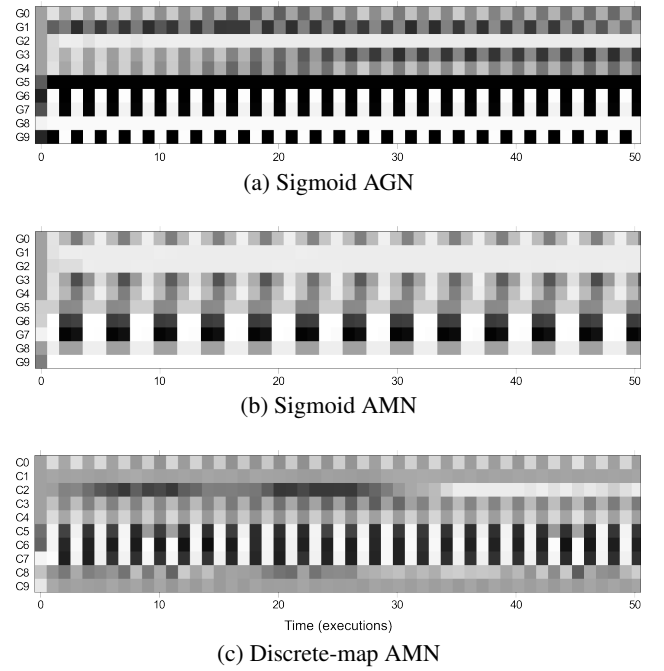


Figure 5: Time series plots of ABNs generating quadrupedal gaits. Actuator angles are input via the first four gene expression levels (G0–G3) or chemical concentrations (C0–C3), and new torque settings are read from the last four (G6–G9, C6–C9). White represents 0, black represents 1, greyscales represent intermediate values.

$$p(n) = \frac{d_n}{t_n} \left( 2 \frac{\eta_b(e_n, e_{n+1})}{\pi} - 1 \right) \left( 1 - \frac{\eta_w(e_n)}{\pi} \right) \sigma_n \quad (3)$$

where  $d_n$  is the distance travelled during epoch  $n$ ,  $t_n$  is the duration of epoch  $n$ ,  $\eta_b$  is the difference in mean heading between two epochs,  $\eta_w$  is the difference in heading within an epoch (as measured during the first and last 50 time-steps of the epoch), and  $\sigma_n$  is a penalty for non-movement: equal to 1 if the robot has not moved for 100 subsequent ABN updates in epoch  $n$ , and 0 otherwise.

In effect, progress is the mean velocity in the required direction, with penalties for turning during an epoch and for non-movement. Assuming movement in a straight-line and no stopping, fitness is equivalent to the expected distance covered during an epoch in the forward or backward direction, whichever is shortest.

**Experimental Settings** A population of 500 is used for this task, to reflect its greater difficulty. In addition to the four actuator angles, the ABN also receives a direction input. This has the value 0 during even-numbered epochs and 1 during odd-numbered epochs. In addition to delivering this signal with the actuator angles, for AGNs and CABNs we also look at the effect of delivering the signal separately through the first regulatory input of one or more genes.

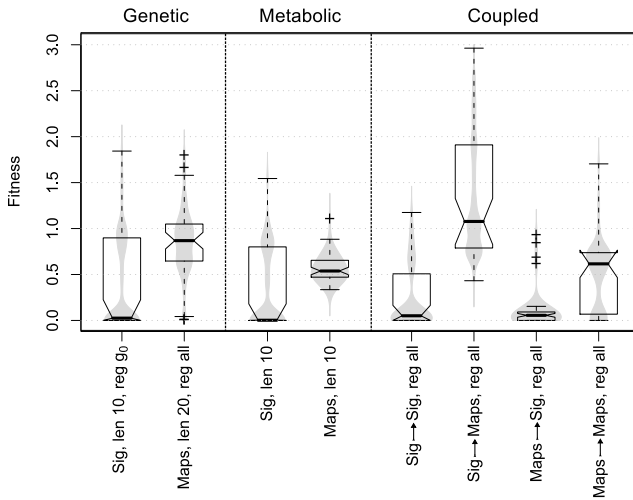


Figure 6: Controlling direction and movement of legged robots. For each function set (or pair of function sets in the case of the coupled network), results for the best-performing combination of solution size and (for genetic and coupled networks) regulatory signal destination are shown. For the latter,  $g_0$  indicates that the control signal was delivered as a regulatory input to the first gene, all indicates that the control signal was delivered as a regulatory input to all genes.

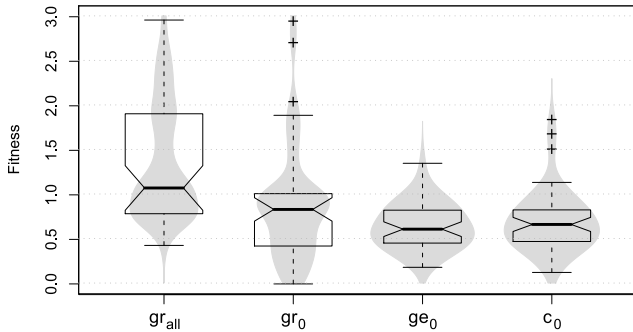


Figure 7: Comparing the effect of delivering the direction signal to different locations within the Sig  $\mapsto$  Maps coupled network.  $gr_{all}$  indicates a regulatory input to all genes,  $gr_0$  is a regulatory input to the first gene,  $ge_0$  is the initial expression of the first gene, and  $c_0$  is the concentration of the first chemical.

**Results** Well-behaved controllers (i.e. those which correctly respond to the direction signal and produce effective gaits) generally have a fitness greater than about 1.5: those with lower fitnesses tend to have periodic or inconsistent behaviours.

Figure 6 compares the fitness distributions of evolved controllers, suggesting that most combinations of ABN model and function set choice do not lead to well-behaved controllers. In fact, the majority of evolved Sigmoidal AGN and AMN were only capable of movement in one direction, giving them a median fitness of zero. Discrete-map AGNs and AMNs achieved higher fitness, but generally did not respond to the direction signal, displaying a range of periodic

Table 2: Occurrence of discrete maps within final solutions from all Sig  $\mapsto$  Maps CABN runs where fitness is greater than 1.5.

Maps	In solutions	Mean occurrences per solution
Baker’s map	100%	2.3
Tunable standard map	78%	1.6
Standard map	78%	1.6
Tunable logistic map	72%	1.2
Arnold’s cat map	61%	1.5
Logistic map	50%	1.7

and aperiodic behaviours.

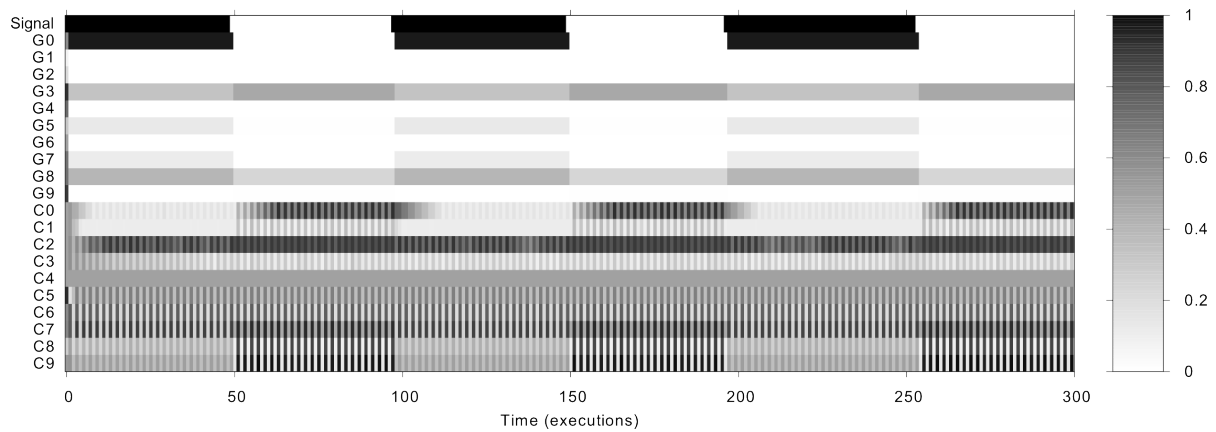
Notably, only coupled networks comprising a Sigmoidal AGN and a discrete-map AMN (denoted Sig  $\mapsto$  Maps) were able to consistently generate competent controllers<sup>1</sup>, and only when the direction signal was delivered as a regulatory input to each gene. Figure 7 shows the effect of delivering this signal to other locations within the Sig  $\mapsto$  Maps coupled networks; showing that delivering the direction signal via a gene’s initial expression or a chemical’s initial concentration was generally ineffective.

Figure 8 shows some representative examples of how these Sig  $\mapsto$  Maps networks control gait and respond to the direction signal. In most evolved networks, the AMN is responsible for generating appropriate patterns of actuator movements and the AGN is responsible for switching between different patterns by regulating the influence of different enzymes. It is interesting to note that their behaviour over time resemble the dynamics of biological biochemical networks, in that a slow-changing genetic network controls a fast changing metabolic network. This may also explain why Sigmoidal functions, which are more amenable to producing slow-changing dynamics, play a productive role within coupled controllers but not within the stand-alone AMN and AGN controllers.

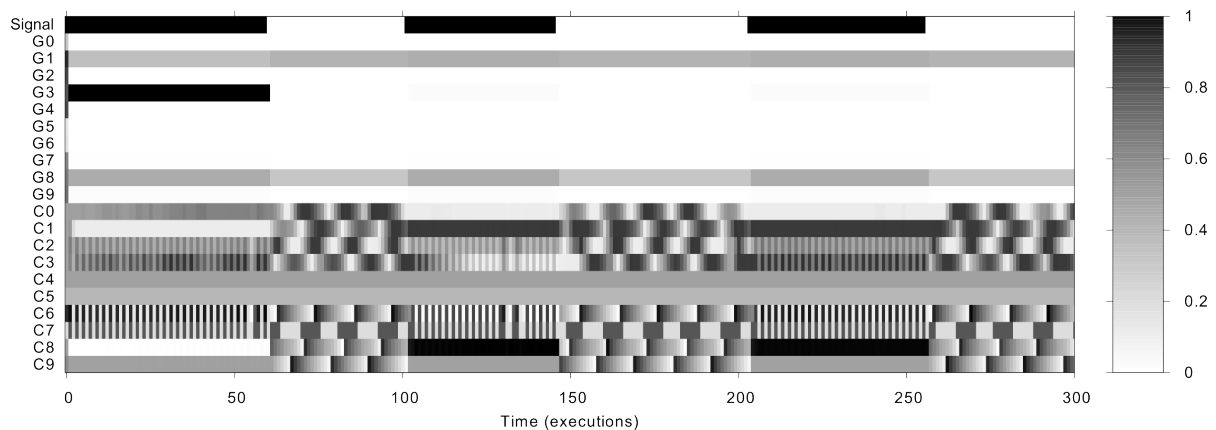
We can hypothesise that there are two reasons why discrete maps are useful for this task. First, they can individually carry out behaviours which would require a number of interconnected Sigmoids to implement—to use a biological analogy, they are the equivalent of a whole biochemical pathway. Arguably, this entails that certain pattern generators can be evolved more readily than in a Sigmoidal network, and using fewer genes. Second, all the discrete maps we use have chaotic phases. When in this phase, their dynamics are highly sensitive to small perturbations, meaning that relatively small changes in gene expression can lead to rapid switching between different attractor states—precisely the behaviour we are looking for in many control tasks.

Table 2 lists the relative occurrence of the different dis-

<sup>1</sup> 18 of the 50 runs generated solutions with fitness greater than 1.5, compared to only a handful for all the other ABNs.



(a) In this example, the AMN generates a single cyclic pattern (C5) which is then scaled and propagated to the outputs (C6–C9). The scaling for each output (and hence the direction of the resulting gait) is determined by the current gene expression pattern.



(b) In this second example, the AMN generates two different cyclic patterns (bunny hopping and a four-legged wading movement), which the AGN switches between in response to changes in the direction signal.

Figure 8: Time series plots of Sig  $\mapsto$  Maps coupled ABNs controlling the direction and gait of a legged robot. The Signal input specifies the required direction of movement. GO–G9 are the expression levels of the genes in the AGN. C0–C9 are the concentration levels of the chemicals in the AMN.

crete maps in the final solutions of successful runs. All of the maps are used by evolution, with most of them appearing in the majority of solutions. The baker's map, in particular, appears in all of the successful controllers, and usually occurs multiple times in these solutions. Since the baker's map is a model of deterministic chaos, this supports our hypothesis that chaotic dynamics are useful. The standard map is also well-represented in evolved solutions, perhaps reflecting its relatively high degree of expressiveness and configurability. It is also notable that the tunable versions of the logistic and Chirikov's maps are often used.

## Conclusions

In this paper, we have shown that artificial biochemical networks can be evolved to control the locomotion of a simulated legged robot. We used two artificial biochemical network models—an artificial genetic network and an artificial metabolic network—and looked at how these models can be

used both individually and when coupled together.

For a simple movement task, where the robot was required to move as far as possible from its starting position, both individual and coupled networks could be evolved to generate suitable gaits. However, for a harder task, where the robot was required to reverse its direction of movement when given a signal, only coupled networks could be evolved to express suitable behaviours. Analysis of the resulting controllers suggests there is a clear separation of effort, with the artificial metabolic network generating patterns of actuator movements and the artificial genetic network switching between different patterns as appropriate.

We found that non-linear discrete maps play an important role in solving the harder of the two problems. When used as functional elements within artificial biochemical networks, these maps provide a useful source of configurable pre-packaged dynamics. Of the maps used in this study, the chaotic baker's map occurred most within evolved solutions.

This finding supports the idea that the inherent instability of chaotic maps makes them useful for rapidly switching between different behaviours.

We also found that the destination of the direction signal has a large effect upon the ability of the networks to solve the harder task. This may reflect the important role that signal recruitment plays within the evolution of biological biochemical networks. Rather than pre-specifying the destination of signals, as we have done in this work, in future work we will look at whether an artificial signalling network can be used to deliver signals to appropriate parts of the genetic and metabolic networks.

## Acknowledgements

This research is funded by the EPSRC (ref: EP/F060041/1). The authors would also like to acknowledge the support of the White Rose Grid in providing computational resources.

## References

- Arnold, V. and Avez, A. (1968). *Ergodic problems in classical mechanics*. Benjamin, New York.
- Banzhaf, W. (2003). Artificial regulatory networks and genetic programming. In Riolo, R. L. and Worzel, B., editors, *Genetic Programming Theory and Practice*, chapter 4, pages 43–62. Kluwer.
- Banzhaf, W. (2004). Artificial chemistries—towards constructive dynamical systems. *Solid State Phenomena*, 97/98:43–50.
- Banzhaf, W. and Lasarczyk, C. (2005). Genetic programming of an algorithmic chemistry. In Koza, J., O'Reilly, U.-M., Yu, T., Riolo, R., and Worzel, B., editors, *Genetic Programming Theory and Practice II*, pages 175–190. Springer US.
- Beer, R. and Gallagher, J. (1992). Evolving dynamical neural networks for adaptive behavior. *Adaptive Behavior*, 1(1):91–122.
- Branicky, M. S. (2005). Introduction to hybrid systems. In Hristu-Varsakelis, D. and Levine, W., editors, *Handbook of Networked and Embedded Control Systems*. Birkhauser.
- Bray, D. (1995). Protein molecules as computational elements in living cells. *Nature*, 376:307–312.
- Chirikov, B. V. (1969). Research concerning the theory of nonlinear resonance and stochasticity. Technical report, Institute of Nuclear Physics, Novosibirsk.
- Clune, J., Beckmann, B. E., Ofria, C., and Pennock, R. T. (2009). Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. In Tyrrell, A. et al., editors, *Proc. 2009 Congress on Evolutionary Computation (CEC 2009)*. IEEE.
- Decraene, J., Mitchell, G. G., and McMullin, B. (2007). Evolving artificial cell signaling networks: Perspectives and methods. In Dressler, F. and Carreras, I., editors, *Advances in Biologically Inspired Information Systems*, pages 167–186. Springer.
- Fontana, W. (1992). Algorithmic chemistry. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II*, pages 159–210. Addison-Wesley.
- Hornby, G., Takamura, S., Yamamoto, T., and Fujita, M. (2005). Autonomous evolution of dynamic gaits with two quadruped robots. *IEEE Transactions on Robotics*, 21(3):402–410.
- Kamio, S., Mitsuhashi, H., and Iba, H. (2003). Integration of genetic programming and reinforcement learning for real robots. In Cantú-Paz, E. et al., editors, *Proc. 2003 Genetic and Evolutionary Computation Conference (GECCO'03)*, volume 2723 of *LNCS*, pages 470–482, Chicago. Springer-Verlag.
- Kantz, H. and Schreiber, T. (2004). *Nonlinear Time Series Analysis*. Cambridge University Press, 2nd edition.
- Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J Theor Biol*, 22(3):437–467.
- Lones, M. A., Tyrrell, A. M., Stepney, S., and Caves, L. S. (2010). Controlling complex dynamics with artificial biochemical networks. In Esparcia-Alcázar, A. I. et al., editors, *Proc. 2010 European Conference on Genetic Programming (EuroGP 2010)*, volume 6021 of *Lecture Notes in Computer Science*, pages 159–170. Springer Berlin / Heidelberg.
- May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, 261:459–467.
- Nicolau, M., Schoenauer, M., and Banzhaf, W. (2010). Evolving genes to balance a pole. In Esparcia-Alcázar et al., editors, *Proc. 2010 European Conference on Genetic Programming (EuroGP 2010)*, volume 6021 of *Lecture Notes in Computer Science*, pages 196–207. Springer Berlin / Heidelberg.
- Păun, Gh. (2000). Computing with membranes. *Journal of Computer and System Sciences*, 61(1):108–143.
- Reil, T. (1999). Dynamics of gene expression in an artificial genome - implications for biological and artificial ontogeny. In *Proc. 5th European Conference on Artificial Life (ECAL'99)*, volume 1674 of *Lecture Notes in Artificial Intelligence*, pages 457–466. Springer-Verlag.
- Seo, K. and Hyun, S. (2008). Genetic programming based automatic gait generation for quadruped robots. In Keijzer, M. et al., editors, *Proc. 2008 Genetic and Evolutionary Computation Conference (GECCO'08)*, pages 293–294, Atlanta, GA, USA. ACM.
- Silva, C. E. (2008). *Invitation to ergodic theory*. AMS.
- Taylor, T. (2004). A genetic regulatory network-inspired real-time controller for a group of underwater robots. In Groen, F. et al., editors, *Intelligent Autonomous Systems 8 (Proceedings of IAS8)*, pages 403–412, Amsterdam. IOS Press.
- Trefzger, M. A., Kuyucu, T., Miller, J. F., and Tyrrell, A. M. (2010). Image compression of natural images using artificial gene regulatory networks. In *Proc. 2010 Genetic and Evolutionary Computation Conference (GECCO'10)*, Portland, Oregon. ACM.
- Ziegler, J. and Banzhaf, W. (2001). Evolving control metabolisms for a robot. *Artificial Life*, 7:171–190.



# Cognitive conditions to the emergence of sign interpretation in artificial creatures

Angelo Loula<sup>1,2</sup>, Ricardo Gudwin<sup>2</sup> and João Queiroz<sup>3\*</sup>

<sup>1</sup> Informatics Area, Department of Exact Sciences, State University of Feira de Santana (UEFS), Brazil

<sup>2</sup> Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering, State University of Campinas (UNICAMP), Brazil

<sup>3</sup> Institute of Arts and Design, Federal University of Juiz de Fora (UFJF), Brazil  
\*queirozj@pq.cnpq.br

## Abstract

Although the emergence of communication has been the topic of various Artificial Life experiments, the study of underlying representational processes finds little discussion. We have previously differentiated between symbolic and indexical interpretation and proposed that symbolic interpretation may act as a shortcut to cognitive traits already acquired. Here we evaluate conditions of this acquired cognitive trait for the emergence of different modalities of sign interpretation. Results show that symbolic processes may act as a cognitive shortcut to a previous acquired cognitive competence even if minimally functional or initially not available.

## Introduction

Computational simulation approaches, such as Artificial Life experiments, are considered to have an important role in the study and modeling of general semiotic processes (see Christiansen and Kirby, 2003, Noble et al., 2010, Cangelosi and Parisi, 2001, Steels, 2003). Communication, vocabulary, grammar are among the processes that have been studied by this synthetic approach (for a review, see Nolfi and Mirolli, 2010, Wagner et al. 2003). In these experiments, semiotic processes are simulated in a social context, involving multiple agents. The process in focus is not pre-defined, but it rather emerges during and by means of agents' interactions. As the main form of interaction between agents, in most of these synthetic experiments, communication has, particularly, been a significant research subject. It depends on the production of representations (by an utterer) and the interpretation of them (by an interpreter). Nevertheless, we find little discussion around representation processes underlying communication such as the types of representations involved and how they can represent something to the agents. If agents communicate, the underlying representational processes are an essential issue to be addressed.

We have previously modeled the emergence of two different types of representational processes (symbols and indexes) and how they emerge in a community of simulated creatures (Loula et al., 2010a). We proposed that a symbolic interpretation process can act as a cognitive shortcut to a cognitive competence that is hard to acquire. Here we propose to assess further this hypothesis and evaluate

cognitive conditions to the emergence of interpretation processes, varying availability and reliability. We apply the same scenario previously used, which involves empirical constraints from studies of animal communication and also theoretical constraints from Peircean pragmatic theory of signs.

In the next section, we review related work in the context of the emergence of communication in Artificial Life research. Next, we briefly describe the theoretical principles and biological motivations that guided our experimental design. We then describe the experiment involving the emergence of different interpretational processes in communication events. Results are presented next, summarizing previous results and exhibiting news ones on the conditions for the emergence of sign interpretation. We discuss achievements and draw conclusions and future directions, in the end.

## Related work

The simulation of the emergence of communication is the topic of various works, but discussions on the underlying semiotic processes finds little space in such literature. Therefore, we will review two representative works that deals with the emergence of communication that are relevant in the context of this work.

Robots were evolved by de Greeff and Nolfi (2010) to execute a navigation task in which two robots had to exchange places in two target areas. The robots could use wireless sensors for an 'explicit signal' communication or they could use their spatial position as an 'implicit signal'. At the end of an evolution process of neural networks that control the robots, de Greeff and Nolfi (2010) described that the robots were able to use 2 or 3 explicit signals to execute the proposed task, but also used one implicit signal to achieve that. They state that explicit signals codify certain conditions in which the emitter robot finds itself and that the implicit signal is a visual perception of the position of the other robot, and that each signal produces a different reaction. Signals are said to be deictic, dependent of spatial-temporal context, but there was no further discussion on what and how robots representationally interpret such signals.

In an experiment with artificial creatures in a grid world, Cangelosi (2001) simulated the emergence of communication systems to name edible and poisonous mushrooms. He had relied on biological motivations to define a food forage goal for the creatures. He proposed the emergence of different modalities of representations in this experiment on the evolution of communication. To classify communication systems, Cangelosi (2001) differentiated signals, with direct relation with world entities, from symbols, also with relation with world entities but also related to other symbols. In his experiments, neural networks were both evolved and trained in various tasks, and, at the end, a shared communication system emerged, involving signals and symbols, according to Cangelosi. But he did not describe how these signals and symbols were interpreted by the creatures and what they actually represented.

Other works have also studied the emergence of communication traits and the acquisition of vocabulary or language among artificial agents (see Nolfi and Mirolli, 2010, Wagner et al. 2003). Nevertheless, we have not found works that have studied the emergence of different types of interpretations processes and differentiated the interpretation processes that emerged.

### Theoretical and Empirical Constraints

Synthetic experiments such as Artificial Life ones are heavily influenced by theoretical principles and biological motivations, and that such background should be an essential part of any synthetic experiment (Parisi, 2001, see also Noble 1997, Loula et al., 2010b). Theoretical principles and biological motivations act as requirements and constraints during the design of the experiments, and influences modeling on different degrees depending on how it constrains the model being built and what decisions it leaves to the experimenter.

To model the emergence of communication processes based on different types of representation, it is certainly important to look at theoretical models and principles, and also look for biological motivations, and avoid arbitrary or naïve assumptions about the underlying processes.

Sign-mediated processes, such as the interpretation of representations in communicative contexts, show a remarkable variety. A basic typology (and the most fundamental one), proposed by Peirce (1958; see Short 2007), differentiates between iconic, indexical, and symbolic processes. Icons are signs that stand for their objects by a similarity or resemblance, no matter if they show any spatio-temporal physical correlation with an existent object. In this case, a sign refers to an object in virtue of a certain quality which is shared between them. Indexes are signs which refer to their objects due to a direct physical connection between them. Since (in this case) the sign should be determined by the object (e.g. by means of a causal relationship) both must exist as actual events. Spatio-temporal co-variation is the most characteristic property of indexical processes. Symbols are signs that are related to their object through a determinative

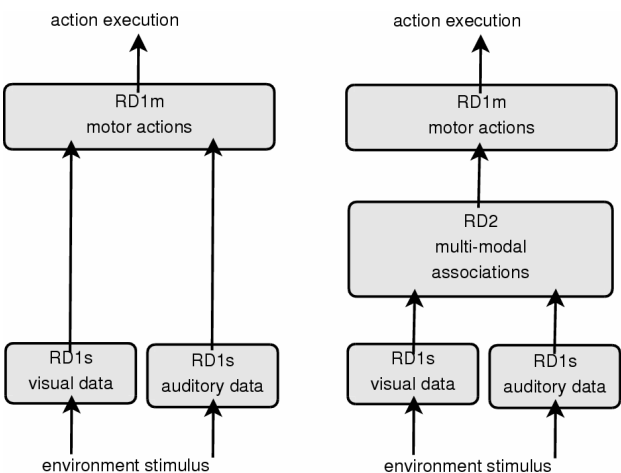


Figure 1: Cognitive architectures for representations interpretations. Left: Type 1 architecture, RD1s are connected directly to RD1m. Right: Type 2 architecture, data from visual RD1s and auditory RD1s can be associated in RD2 before connecting to RD1m.

relation of law, rule or convention<sup>1</sup>. A symbol becomes a sign of some object merely or mainly by the fact that it is used and understood as such by the interpreter, who establishes this connection.

Communication is a process that occurs among natural systems and as such we can employ empirical evidences on building our synthetic experiment. Animals communicate in various situations, from courtship and dominance to predator warning and food calls (see Hauser, 1997). And following Peirce’s definition of symbols, many animals can actually be capable of communicating by means of symbols (Ribeiro et al., 2007).

To further explore the mechanisms behind communication, a minimum brain model can be useful to understand what cognitive resources might be available and process underlining certain behaviors. Queiroz and Ribeiro (2002) described a minimum vertebrate brain for vervet monkeys predator warning vocalization behavior (Seyfarth et al 1980). It was modeled as being composed by three major representational relays or domains: the sensory, the associative and the motor. According to such minimalist design, different first-order sensory representational domains (RD1s) receive unimodal stimuli, which are then associated in a second-order multi-modal representation domain (RD2) so as to elicit symbolic responses to alarm-calls by means of a first-order motor representation domain (RD1m).

Our objective is to model the emergence of indexical and symbolic interpretation competences, so the first step is to specify the requirements for each and also how to recognize each of them in the experiment. Indexical interpretation is a reactive interpretation of signs, such that the interpreter is directed by the sign to recognize its object as something spatio-temporally connected to it, so for our creatures to have

<sup>1</sup> Differently from Cangelosi’s (2001) definition of symbol, based on Deacon’s approach (1997), Peirce (1958) did not require symbols to be related to each other to be called symbols.

this competence, they must be able to reactively respond to sensory stimulus with prompt motor answer. In the minimum brain model, this corresponds to an individual capable of connecting RD1s to RD1m without the need for RD2. But a symbolic interpretation undergoes the mediation of the interpreter to connect the sign to its object, in such a way that a habit (either inborn or acquired) must be present to establish this association. Thus, in symbolic interpretation, RD2 must be present once it is the only domain able to establish connections between different representation modes. Thus, our artificial creatures must be able to receive sensory data, both visual and auditory, in its respective RD1s, that can be connected directly to RD1m, defining motor actions (Type 1 architecture), or connected to RD1m indirectly, through the mediation of RD2, that associates auditory stimulus to visual stimulus acting as an associative memory module (Type 2 architecture) (see figure 1). To evaluate what conditions might elicit each response type – indexical or symbolic –, we implemented these two possible cognitive processing paths as mutually exclusive paths: either the creature responds to auditory events indexically and reactively responds with motor actions, or the creatures responds to auditory events symbolically and associates them with a visual stimulus and responds as if that was really seen. For an external observer, who only watches the information available to the creature and its motor responses, it may not be possible to see changes in the interpretation process. But the underlying mechanisms behind each semiotic process are qualitatively different.

### The experiment

The scenario to test the conditions for the emergence of semiotic processes is inspired by food foraging behavior of animals. One way animals cooperate in such task is by vocalizing for food quality, recruiting other group members to feed. Inspired by such biological motivation, we simulate a scenario of artificial creatures evolved to collect resources in a virtual environment.

Lower quality resources are scattered throughout the environment and a single location receives highest quality resources. One creature (vocalizer) is placed in this high quality resource position, vocalizing a sign continuously. At the start of simulation, the other creatures (interpreters) do not know how to respond appropriately to sensory inputs and neither recognizes the sign vocalized as a sign. But an evolutionary process of variation and selection is applied, allowing the evolution of individuals to better accomplish the task of resource foraging. During the evolutionary process, for each start-up conditions, we observe the emergence of indexical or symbolic interpretation for the vocalizations.

The environment is a 50 by 50 grid world (figure 2) and there are 20 random positions with only one resource unit each. There is also one position with 500 resource units, where an immovable vocalizer creature is placed. The vocalizer's sole behavior is to produce a single vocal sign, reproduced at every instant. Fifty interpreter creatures are randomly placed in this grid.

Interpreter creatures are capable of visually sensing food up to a distance of 4 positions and sensing vocalizations up to a distance of 25 positions. This sensory range difference models

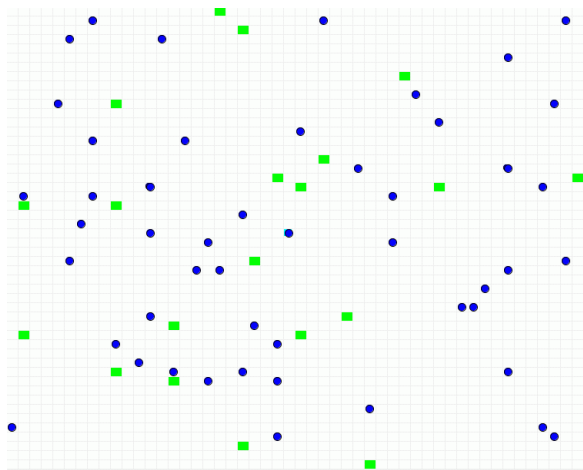


Figure 2: The grid environment. Creatures are blue circles, low quality resource positions are in green cells, and high quality one in the cyan cell in the center.

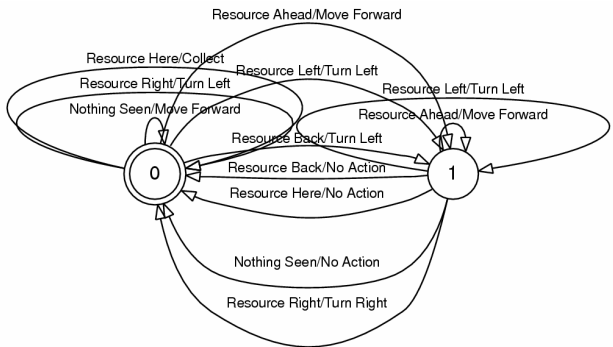


Figure 3: State diagram of a sample FSM that controls the creatures. The circles are states and a double circle marks the start state. Arches represent transitions and are labeled according to the sensory event and the action to take over when that event occurs. This FSM has only visual inputs and 2 states to simplify the diagram, but there can be more arches for auditory inputs (vocalization and its position) and up to 4 states.

an environment where vision is limited by the presence of other elements such as vegetation, restraining far vision such as in an open field. These creatures can either see a resource and its position or hear a vocalization and its position, if any of them is within range.

Interpreters are controlled by finite state machines (FSM), with up to 4 states (see figure 3). An FSM was chosen as the control architecture because the analysis of how it is functioning is quite simple and direct, permitting direct identification of the processes underlying the creatures' cognition. Input events to the FSM include 5 visual events for resource in 5 different positions (ahead, left, right, back, or same position), 4 auditory events for vocalization in 4 positions (ahead, left, right, back), and 1 event for nothing seen or heard. Outputs from FSM can be one of the 5 motor

actions for creatures: move forward, turn left, turn right, collect resource, or do nothing.

The creatures can respond to visual inputs with one of the motor actions, and can also respond to auditory input with a direct motor action (a reactive, indexical process) (Type 1 architecture). Alternatively, before an input is sent to the FSM, they can also choose to establish an internal association between the heard stimulus and the visual representation domain (Type 2 architecture). This internal association links what is heard with the view of a collectible resource, i.e. the creature can interpret the sign heard as a resource and act as if the resource was seen. As a result, an auditory input is exchanged by an equivalent visual input and the FSM is executed with that input. Additionally, the creature may also ignore the sign heard, interpreting it as nothing and acting as if no sensory data was received.

At start, creatures are controlled by randomly constructed FSMs, and are all placed at random in the environment. They are allowed to collect resources for 10 trials of 100 iterations each trial. Creatures collect resources by executing the specific action, removing one unit from the resource at each time step. When no more units are available at the resource, it disappears.

### Creatures evolution

At the all trials, the 10 best creatures in the foraging task (those that collected the most resource units) are selected to create next generation. These 10 individuals are copied to the next population and the 40 remaining individuals are a product of mutations and crossovers of the FSMs of the best individuals.

The mutations can be of changing an action in transition, changing the next state after a transition, changing the start state, add a state and remove a state. There can also be a mutation of the cognitive architecture type, as described below. The number of mutations is selected from a Poisson probability distribution with an expected value of 3. The crossover has a 50% chance of occurring and it exchanges states and transitions originating from the selected states between two FSM in a uniform way. All FSM undergo a correction process to fix error that might occur during these operations, such as a transition pointing to a non-existing state.

The experiment runs for 500 generations, normally with two distinct moments. In the first 200 generations (cycle 1), the vocalizer creature is not present and interpreters do not have an auditory sensor, but this first cycle will be omitted in one of the simulation scenarios. In the 300 subsequent generations the vocalizer creature is present and interpreters are able to hear (cycle 2).

At the start of cycle 2, all creatures are set to ignore the vocalizations, as if it was not relevant, however, there is also a small mutation probability for changing the type of response to vocalizations. These can be of reacting to them by moving towards the resource, or to linking it with the view of a resource. This corresponds to a change to a Type 1 cognitive architecture (indexical) or to a Type 2 cognitive architecture

(symbolic). The probability of going from Type 1 architecture to Type 2 architecture is lower than the other way around to simulate the fact that such a significant cognitive change is not that easy to happen.

We expect that creatures adapt to the foraging task by responding to the auditory input of vocalizations. Since they can not see the high quality resource position, they must rely on the vocalization to guide their movements in this direction. We are interested in observing the overall adaptation process to the foraging task, and are specially focused on the type of interpretation process, related to the cognitive architecture type, that might result.

## Results

In previous work, we have run two initial experiments to evaluate the emergence of either an indexical interpretation or a symbolic interpretation of vocalizations (Loula et al., 2010a). Such experiments involved 2 cycles as described above, varying the way motor actions needed to be coordinated. In the first experiment, creatures just had to have the specified action as output of the FSM to execute that action. In this scenario, we observed that indexical interpretation was the competence acquired by creatures to deal with communication, with direct association between auditory signs and motor actions. But in a second experiment, for motor actions to be executed, the creatures needed to first output a null action before any movement action, that way it would be harder to learn motor coordination. In this alternative scenario, symbolic interpretation was the emerging competence, instead of an indexical one like it happened in the previous case. We made the hypothesis that acquiring symbolic competence would act as a *cognitive shortcut*, by reusing a previously acquired ability in cycle 1: to appropriately respond to visual data with motor actions. We proposed that a symbolic interpretation process can happen if a cognitive trait is hard to be acquired and the symbolic interpretation of a sign will connect it with another sign for which the creature already has an appropriate response.

### Single cycle scenario

In face of the fact that there should be a previously acquired competence for symbolic interpretation to benefit from, a subsequent question is to ask what would happen to sign interpretation if such previous competence is not present. If the creature does not respond in a proper manner to visual input, a cognitive shortcut to this uncoordinated competence would not help the foraging success. As cycle 1 acts as a first step in which creatures are dedicated to learn visual-motor coordination, we removed this cycle in a new scenario, in which the simulation begins in cycle 2 with the vocalizer at the center and interpreter creatures able to hear but starting with random FSMs. The need to first output a null action before any movement action remains, so it is hard to learn motor coordination. Figure 4 shows the performance of creatures in foraging and the type of interpretation used.

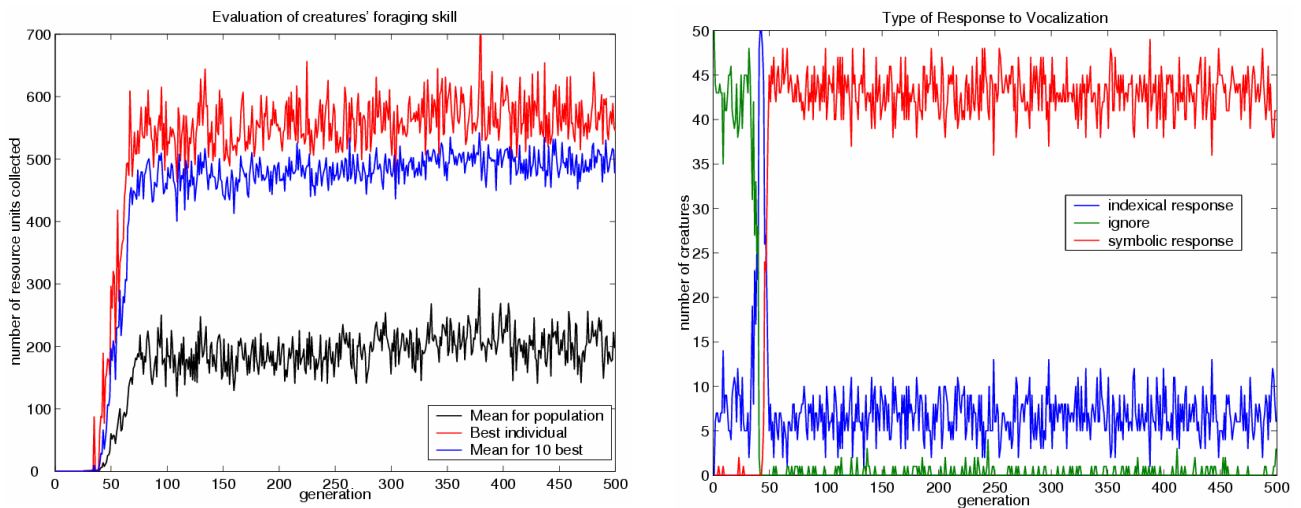


Figure 4: Evaluation of foraging task and type of response to vocalizations along the generations for the one cycle only experiment.

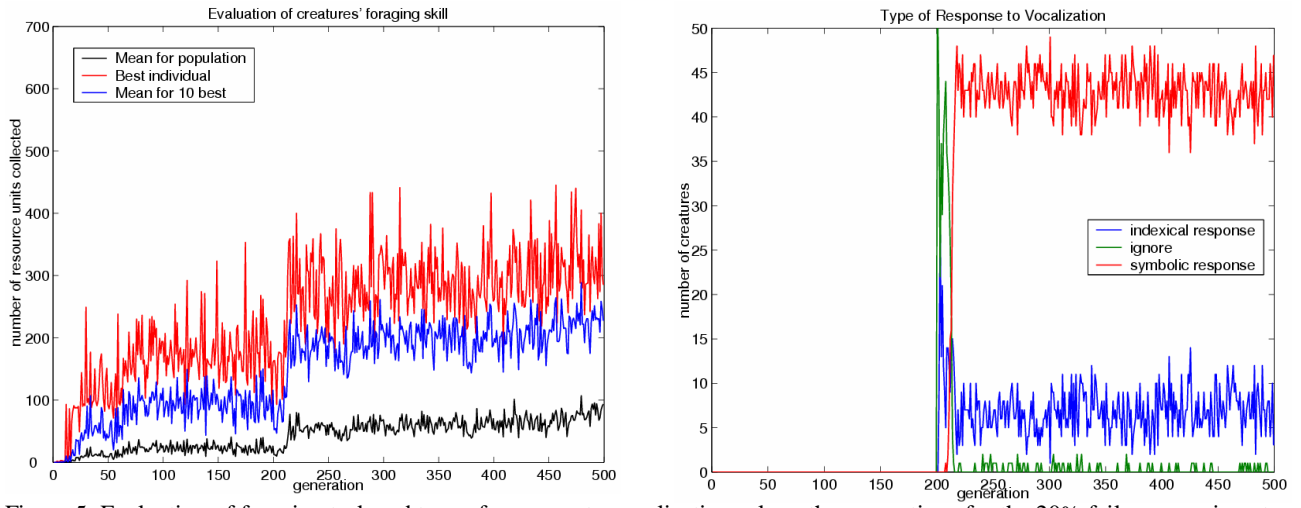


Figure 5: Evaluation of foraging task and type of response to vocalizations along the generations for the 20% failure experiment.

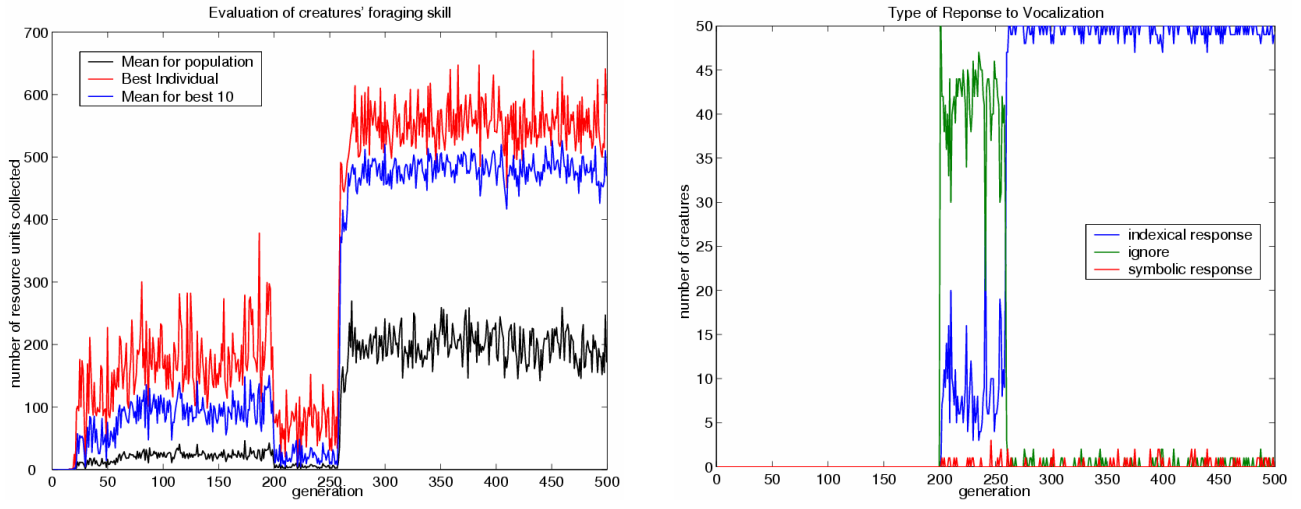


Figure 6: Evaluation of foraging task and type of response to vocalizations along the generations for the 50% failure experiment.

As we can see from the graphs, the experiment had three phases. At first, no resources were collected and creatures opted to ignore signs produced by the vocalizer. Then there was a transition phase, where the amounts of resources increased rapidly along generations and creatures gave up ignoring signs and started an indexical interpretation of them. Then creatures turn to a symbolic interpretation of sign and the amount of resources collected further increases and then stabilizes. To better understand what happened in such transitions, the FSMs of creatures have to be further detailed.

From the first generation until generation 25, creatures did not demonstrate any motor coordination and were not able to collect resources, and *most creatures just ignored signs*. In generation 26, one creature was able to move forward and collect when a resource was in front of it, but it still ignored signs.

This remained the same until generation 39, where one creature was able to turn right when a resource was seen at right side, and this creature was also *responding indexically to signs*, by going towards the vocalizer when a sign was heard in front of it. By generation 40, half of the population is interpreting signs indexically and the other half is ignoring it. Most of the creatures could move towards a seen resource, but there were still some useless outputs from the FSM, state/transition combinations that would make a creature stop responding effectively, and they still would not move when nothing is seen.

At generation 44, one creature starts *interpreting signs as symbols*, relating the sign heard with the view of a resource, able to collect 67 resource units, while the best performing creature collected 77, but interpreting signs as index. Nevertheless, creatures still had problems in motor coordination. By generation 46, half of the creatures were symbolically interpreting signs, and by generation 50, almost all of them did so. From there on, all 10 best performing creatures used symbolic interpretations (and most of the others too), and the number of collected resources increased rapidly as creatures acquired a best performing FSM, that would always respond effectively to the inputs received.

So even though, there was no cycle dedicated to acquire a previous competence that could be re-used by a symbolic interpretation, the evolution process allowed first for visual-motor coordination to appear before sign interpretation (either as index or symbol) started. Thus, there was at least a little visual-motor competence to be re-used by symbolic interpretation.

### Cognitive module malfunction scenario

To further evaluate the way symbolic interpretation acts as a cognitive shortcut, we set up one more scenario. Since there is re-use of a previous acquired cognitive competence, we tested how reliable should this competence be for this new symbolic process to connect to it. The scenario is similar to the one above, but we brought back cycle 1 before cycle 2, so the creatures had time to acquire visual-motor coordination. However, in this reliability test, we introduced a failure chance in the visual-motor coordination after cycle 1, simulating a malfunctioning cognitive module. Given an output from the FSM in response to a visual input, this output (an action) would have a chance of changing to a different

one. If the input is 'Resource Left' and the output from the FSM is 'Turn Left', for example, it could be changed to 'Turn Right'. Outputs that are responses to auditory inputs are not subject to such changes. This way visual-motor coordination would be defective and processes relying on it would be jeopardized.

The first simulation of this malfunctioning in visual-motor coordination applied a 20% chance of output change. The results are presented in figure 5. Compared to a previous experiment with 2 cycles but no malfunction (Loula et al., 2010a), it is possible to notice that the number of collected resources during cycle 1 is similar in both experiments, but in the second cycle it is quite different: while in the previous experiment the best creature collected between 500 and 600 units, in this unreliable module experiment, the best creature collect only around 300 units. This shows that the foraging efficiency has dropped down with the imposed malfunctioning. Looking at the type of response, signs ended up having a *symbolic interpretation*, thus the unreliable visual-motor connections were in fact reused, despite the fact that it was not an efficient module. Comparing with the cited previous experiment, the interpretation type graph is quite similar.

Taking a closer look at simulation outcome, results show that from generation 200 to 210 the foraging performance did not improve. Initially creatures ignored signs, but by generation 202, a few creatures start trying to respond to signs in an indexical manner. These creatures with type 1 architecture, nevertheless, are not able to move towards the vocalizer and still rely in the defective visual-motor coordination. In generation 208, almost all creatures are ignoring signs again.

By generation 210, a symbolic interpreter appeared and it was able to collect more than 200 resource units. Even though visual-motor coordination was degraded, it still performed better than the random actions of a creature trying an indexical response. From this generation on, the number of creatures *interpreting signs symbolically* increased and, by generation 218, almost all creatures followed this type of interpretation.

To further test the effects of a malfunctioning of a cognitive module, the chance of changing actions was increased to 50% in a new simulation, with the expected effect of turning the visual-motor coordination so unreliable that its reuse would be not be possible. Results of this simulation are shown in figure 6.

In this new simulation run, we observe that after cycle 1 the number of collected resources dropped considerably more, to about half of the amount at the end of cycle 1. This was expected since the creatures are using a quite defective control model that is not able to cope with the task of foraging resources efficiently anymore.

Until around generation 250, creatures had this bad performance, but in the meanwhile sign interpretation was varying from ignoring signs to indexical response. The best performing creatures were most ignoring sign, though, indicating that indexical interpreters were not able to successfully respond to signs. One or two creatures with symbolic response were created but disappear right after as its performance was not consistent, due to the dependence on visual-motor coordination.

At generation 258, there appeared a creature with *indexical interpretation*, able to collect 210 resource units. This creature was able to effectively respond to signs by going towards the vocalizer when it was located ahead or to the left. Therefore, this indexical interpreter was able to rely on a direct connection between auditory input and motor actions, and avoided using the faulty cognitive module. The number of collected resource units along generations increased fast, and the best creature (an indexical interpreter) on generation 270 was collecting almost 600 resource units and this performance was consistently kept until the end of simulation. Notice that if we compare the efficiency of creatures in the 20% failure chance simulation with this 50% failure chance simulation, it is clear that even though the second simulation had a worst damage to the visual-motor module, it was able to achieve better performance at the end.

## Discussion

In this paper, we continue investigating conditions for qualitative different interpretation processes to emerge in a communicational context. Previous results showed that symbolic interpretation can emerge when the appropriate motor coordination is a hard skill to acquire, and therefore symbolic processes can act as a cognitive shortcut, mapping auditory signs to visual input and reusing visual module mapping to motor actions. Here we test other conditions for this cognitive shortcut to be established.

First we removed the first cycle, when creatures were allowed to acquire visual-motor coordination, which could be reused through a cognitive shortcut. Consequently, adequate auditory and visual responses needed to be acquired at the same time. From this single cycle experiment, it is possible to observe that even though the vocalizer and the hearing sensor were available from start, creatures did not use signs at all in a first moment. It was necessary to first have minimum visual-motor coordination for signs to start being interpreted by creatures. Indexical interpretation was the first attempt as a response to signs. As trying to acquire visual-motor coordination and also a sign-motor coordination is a tough route, the symbolic interpretation diminished this effort and became the dominant strategy.

To further evaluate the cognitive shortcut stability, we imposed a variable malfunctioning to the visual-motor connections. At first, a 20% of changing actions specified by the visual module still conducted to the establishment of symbolic processes, with reuse of a degraded module, but that still allowed a relative increase in foraging efficiency. A higher failure of 50% proved to worsen the performance of the visual control module considerably more, and allowed indexical interpretation of sign to be established, as a way to avoid reusing it. And, even though symbolic processes were established in the 20% failure scenario, it seems that creatures got trapped in a 'local maximum' performance, as the foraging efficacy of creatures in the 50% failure scenario was much better.

## Conclusion

Communication necessarily involves an utterer, who produces a sign, conveyed to an interpreter, in whom the sign produces its effect. And signs can be of different types according to the way it is connected during interpretation process to its referent. We proposed that, for two types of signs – indexes and symbols – to be interpreted, different cognitive paths had to be followed, one with direct mapping of signs to motor actions (indexical interpretation) and another one with a mapping of signs into another representation form (symbolic interpretation) and then to motor actions.

We proposed that a cognitive shortcut can be established by symbolic interpretation processes, by establishing bridges to reuse previous acquired competences. We confirmed here that the cognitive module to which the symbolic interpretation is connecting to must be already established, otherwise there is no advantage in such connection. But it does need to be fully functional, as minimal visual-motor coordination is sufficient to begin a symbolic interpretation process, according to the single cycle experiment, and even a moderately damaged module can also be reused as a cognitive shortcut.

Even further investigations on differentiating indexical and symbolic processes have to be done. Other aspects and conditions should be tested to better understand what leads sign interpreters to each of them, for example, how can an agent handle both of them at the same time or how does other cognitive competences influence this process. We expect that the discrimination of these semiotic processes and the cognitive apparatus necessary for each of them will bring forth more discussion on representation process in experiments on the emergence of communication and language.

## References

- Cangelosi, A. (2001) Evolution of communication and language using signals, symbols and words. *IEEE Transactions on Evolutionary Computation* 5(2), 93-101
- Cangelosi, A. and Parisi, D. (2001) Computer simulation: A new scientific approach to the study of language evolution. In Cangelosi, A. and Parisi, D., editors, *Simulating the Evolution of Language*, chapter 1, pages 3–28. Springer Verlag.
- Christiansen, M. H. and Kirby, S. (2003) Language evolution: consensus and controversies. *Trends in Cognitive Sciences*, 7(7):300–307.
- de Greeff, J. and Nolfi, S. (2010) Evolution of implicit and explicit communication in mobile robots. In *Evolution of Communication and Language in Embodied Agents*, pages 179–214. Springer Verlag.
- Deacon, T. (1998) *The Symbolic Species: The Co-Evolution of Language and the Brain*. W. W. Norton & Company, New York.
- Hauser, M. D. (1997) *The Evolution of Communication*. MIT Press, Cambridge, MA.
- Loula, A., Gudwin, R., and Queiroz, J. (2010a) On the emergence of indexical and symbolic interpretation in artificial creatures, or What is this I hear? In Fellermann, H., et al., editors, *Artificial Life XII*, pages 862–868. MIT Press.
- Loula, A., Gudwin, R., Ribeiro, S., and Queiroz, J. (2010b) On Building Meaning: a biologically-inspired experiment on symbol-based communication. In Cutsuridis, V., Hussain, A., Barros, A.K., Aleksander, I., Smith, L., and Chrisley, L., editors, *Brain Inspired Cognitive Systems*, pages 77-94. USA: Springer.
- Noble, J. (1997) The scientific status of artificial life. In *Fourth European Conference on Artificial Life (ECAL97)*, pages 28-31. MIT Press/Bradford Books.



- Noble, J., De Ruiter, J.P., and Arnold, K. (2010) From monkey alarm calls to human language: How simulations can fill the gap. *Adaptive Behavior* 18(1):66–82.
- Nolfi, S. and Mirolli, M. (2010) *Evolution of Communication and Language in Embodied Agents*. Springer.
- Parisi, D. (2001) *Simulazioni - la realtà rifatta nel computer*. Il Mulino, Bologna.
- Peirce, C.S. (1958) *Collected Papers of Charles Sanders Peirce*. Harvard University Press, Cambridge, Mass.
- Queiroz, J. and Ribeiro, S. (2002) The biological substrate of icons, indexes, and symbols in animal communication: A neurosemiotic analysis of vervet monkey alarm calls. In Shapiro, M., editor, *The Peirce Seminar Papers* 5, pages 69–78. Berghahn Books, New York.
- Ribeiro S., Loula, A., Araújo I., Gudwin, R., and Queiroz, J. (2007) Symbols are not uniquely human. *Biosystems* 90:263-272.
- Seyfarth, R., Cheney, D., and Marler, P. (1980). Monkey responses to three different alarm calls: Evidence of predator classification and semantic communication. *Science*, 210, 801–803.
- Short, T.L. (2007) *Peirce's Theory of Signs*. Cambridge University Press.
- Steels, L. (2003) Evolving grounded communication for robots. *TRENDS in Cognitive Sciences*, 7(7): 308-312.
- Wagner, K., Reggia, J. A., Uriagereka, J., and Wilkinson, G. S. (2003) Progress in the simulation of emergent communication and language. *Adaptive Behavior*, 11(1):37-69.

# Evolutionary acquisition of genetic program for death

Tadao Maekawa<sup>1</sup>, Osamu Ueno<sup>2</sup>, Norie Kawai<sup>3</sup>, Emi Nishina<sup>4</sup>, Manabu Honda<sup>2</sup> and Tsutomu Oohashi<sup>3</sup>

<sup>1</sup>Faculty of Environmental and Information Sciences, Yokkaichi University

<sup>2</sup>Department of Function Brain Research, National Center of Neurology and Psychiatry

<sup>3</sup>Department of Research and Development, Foundation for Advancement of International Science

<sup>4</sup>Center of ICT and Distance Education, the Open University of Japan

## Abstract

As part of our research on <programmed self-decomposition>, we formed the hypothesis that originally immortal terrestrial organisms evolve into ones that are programmed for autonomous death. We then conducted simulation experiments in which we examined this hypothesis using an artificial ecosystem that we designed to refer to a terrestrial ecosystem endowed with Artificial Chemistry (ACChem). Our findings suggest that, in the case of a mortal organism appearing among a population of immortal organisms as a mutant which evolutionarily acquires a genetic program for death by means of self-decomposition, this organism and its surviving offspring surpass immortal organisms and eventually prosper with adaptive divergence under various environmental conditions within a certain probability.

## Introduction

We modeled autonomous death, which is the significant and universal attribute of terrestrial life, as <programmed self-decomposition> (Oohashi, et al. 1987, 2009). Our research has proceeded through a series of studies that look into the existence of autonomous death by means of experiments in the field of molecular cell biology with existing living organisms as subjects; concurrently, by means of evolutionary simulations of Artificial Life (ALife), we raise the possibility that mortal organisms having autonomous death are superior to immortal organisms (Oohashi, et al. 1987, 1996, 1999, 2001, 2009, 2011).

Throughout this study, we take note of the fact that mortal organisms endowed with programmed self-decomposition are more complex than immortal organisms in both structural and functional aspects, and that the former can better increase the prosperity of their offspring than the latter can. Therefore, we formed the hypothesis that [originally immortal terrestrial organisms evolved into ones capable of autonomous death.] We then conducted a preliminary investigation using an artificial ecosystem SIVA-III (Oohashi et al. 1996) of our own design and obtained results that suggest the robustness of our hypothesis (Oohashi et al. 2001).

Thereupon, we constructed a more sophisticated model for a more detailed investigation making use of an artificial <ACChem> ecosystem SIVA-T05. The essential questions we sought to answer are as follows: Would an individual mortal organism, overwhelmed by immortal organisms, become extinct, or could such an individual survive and produce

offspring? If it survived and produced offspring, what kind of power relationships would be established between such mortal organisms and the immortal ones?

Our findings suggest that a mortal organism, born among a population of immortal organisms, cannot reproduce and becomes extinct in many cases. Nonetheless, a number of mortal organisms did manage to survive at a small but significant rate. Moreover, once a mortal organism survives, it extends its habitation area, surpasses immortal organisms and prospers without exception. This paper provides details of the above findings.

## Methods

### 1) Programmed Self-Decomposition Model

We previously designed “Programmed Self-Decomposition (PSD) Model” (Oohashi, et al. 1987, 2009) based on a hypothesis concerning death universally observed in terrestrial life. This hypothesis is summarized below since it constitutes the framework of the current study, which examines the acquisition of death. The terrestrial ecosystem forms a nearly closed system in that both its space and substance are limited. Accordingly, to maintain the stability of terrestrial life activities, the space and substance of the environment used by life activities have to be returned to the environment. That is to say, the ecosystem must return to its original state. The mechanism for restoring the terrestrial ecosystem has conventionally been explained by the principle of biological circulation called the food chain (Odum, 1971), which is a biomolecular recycling mechanism for terrestrial life. We set forth a new hypothesis complementary to that of the food chain. In our view of the terrestrial ecosystem, besides the restoration of the environment due to the food chain, another hidden mechanism is fundamentally built into every life individual, by which it autonomously decomposes itself so as to contribute to the restoration of the environment. We regard the phenomenon of decomposition based on the life individual’s own effort, called self-decomposition, to be a controlled biochemical process of returning the substance and space that that individual possesses to the environment for the purpose of restoring the environment to its original state. We call this programmed self-decomposition (PSD) (Oohashi, et al. 1987, 2009). We posit that the effect of the mechanism of self-decomposition does not directly accord benefits to the

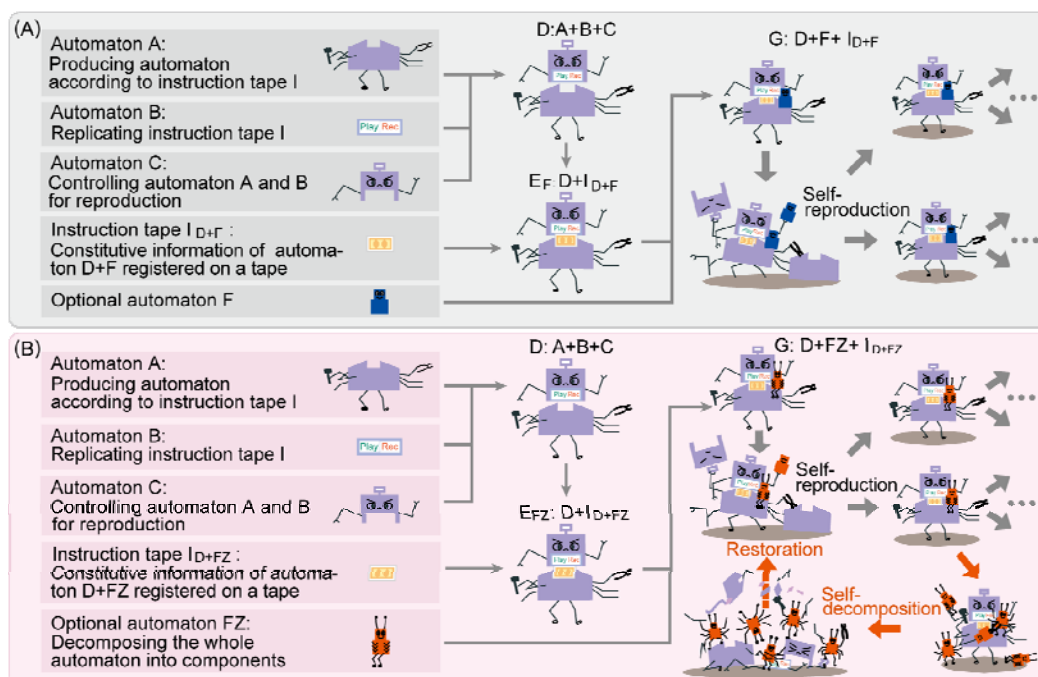


Figure 1. Von Neumann's self-reproductive automaton and Oohashi's self-reproductive, self-decomposable automaton.

(A) Von Neumann's self-reproductive automaton model. This is an immortal type model without an autonomous mechanism for the restoration of the environment to its original state. (B) Oohashi's self-reproductive, self-decomposable (SRSD) automaton model. This model uses von Neumann's self-reproductive automaton model as its prototype. It has a programmed mechanism contributing to the restoration of the environment to its original state through autonomous individual death with self-decomposition, which is an essential feature of terrestrial life. Two activation modes are defined for the self-decomposition automaton FZ. The first one is activated by a signal input from outside, indicating unconformity between the life and its habitation environment. The second mode constitutes the end of the life span.

decomposing individual itself, but rather it enhances benefit to the species sharing its genetic lineage as well as to the ecosystem as a whole. It is necessary to conduct experiments to determine whether such a phenomenon is evolutionarily selected or not. We have developed a self-reproductive, self-decomposable (SRSD) automaton, on the basis of which we have been examining the PSD model, using von Neumann's self-reproductive automaton model (Neumann, 1951) as a prototype (Figure 1) (Oohashi et al., 1987, 2009).

## 2) Architecture of SIVA-T05

We developed a virtual ecosystem series SIVA (Oohashi, et al. 1996, 2001, 2009) configured with Oohashi's SRSD automaton installed in a finite, heterogeneous environment consisting of virtual biomolecules having chemical reactivity. Since constructing SIVA-III, a pioneering prototype for an AChem system, in 1996 (Oohashi et al. 1996), we have continued to develop SIVA as a virtual ecosystem based on AChem. To promote the main purpose of AChem, namely, the achievement of a closer relationship with existent terrestrial life, SIVA-T05, a new version of SIVA, has been developed to have a biomolecular hierarchy, as put forth in Network Artificial Chemistry (Suzuki, 2004), which is an AChem system that succeeds in simulating molecular conformation and reactivity by arranging the strength of cohesion between elements into a hierarchy. SIVA-T05 was adopted as a simulator in this paper.

**A) Environmental Design of SIVA-T05.** To simulate the characteristics of a terrestrial environment with limited amounts of materials and energy distributed in a finite space, the virtual space of SIVA-T05 is designed to be a two-dimensional lattice consisting of 16 x 16 (= 256) spatial blocks. A single spatial block is defined as 8 x 8 (= 64) pixels for habitation points. One habitation point is occupied by one virtual life individual (VLI) and vice versa [Figure 2(A)]. Environmental conditions can be independently defined for each spatial block, and those of the 64 habitation points in the same spatial block are configured to always be homogeneous. VLIs change the quantity of available substances in the environment by importing them into their bodies as materials for self-reproduction and by exporting them through self-decomposition. Since all VLIs in one spatial block share the same environmental conditions, the population of VLIs in that block significantly affects local conditions. Consequently the divergence of local environmental conditions across the whole ecosystem is gradually emphasized along with the proliferation of VLIs, as would also occur in a terrestrial ecosystem.

The temperature gradient and the initial distribution of virtual energy and four kinds of virtual inorganic biomaterials (see the next section) consisting of VLIs are heterogeneous across the whole ecosystem as shown in Figure 2 (B). No substances other than virtual inorganic biomaterials exist in the initial environment. To simulate the effects of solar energy and its

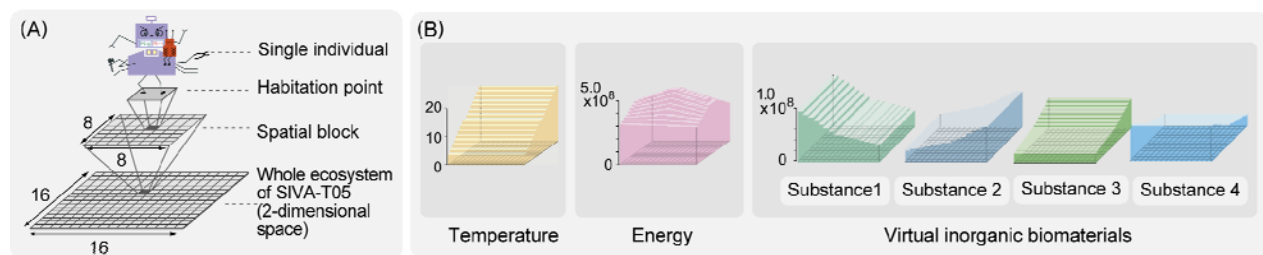


Figure 2. Environmental conditions of the virtual ecosystem SIVA-T05 are designed to be finite and heterogeneous. (A) Spatial design. The virtual space of SIVA-T05 is a two-dimensional lattice (B) Spatial distribution of environmental conditions. Left: Distribution of environmental temperature. Initial distribution of energy stocked in each spatial block. Right: Initial distribution of four kinds of virtual inorganic biomaterials (VI). Each substance flows between adjacent spatial blocks to restore the environment to the initial condition when the amount of a substance goes above or below that of the predetermined level.

diffusion and radiation in the terrestrial ecosystem, a predefined amount of energy per time unit is refilled, and the total amount of energy in each spatial block must not exceed a predetermined threshold. The amount of refilled energy and the upper limit of total energy are set at appropriate levels so that a simulation does not become meaningless, that is, not so small that no VLI can live stably and not so large that all VLIs can always live without any failure.

**B) Design of Virtual Life in SIVA-T05.** In SIVA-T05, we have designed a new type of virtual life based on the hierarchical biomolecular covalent bond (HBCB) model (Oohashi et al. 2009). Table 1 shows the design of the hierarchical structure of virtual biomolecules based on the complexity of the interatomic network of actual biomolecules that compose terrestrial life.

Virtual biological polymers (VPs) and virtual biological monomers (VMs) are categorized into two groups: the functional module group and the constitutive information group, which in terrestrial life correspond to the phenotype and the genotype respectively.

Basically each substance in a certain class consists of several elements belonging to the next lower class. For example, a virtual organic biomaterial (VO) consists of several virtual inorganic biomaterials (VIs), and a VM consists of several VOs. Several VMs constitute a functional unit, which is a subclass of its VP class, and several functional units constitute a larger VP. In the present simulation experiments, we designed five VMs as a single functional unit. A functional unit serves as one word in the SIVA language in the functional module group and also constitutes a virtual codon (Vcodon) in the constitutive information group. Oohashi's SRSD automaton is installed as an artificial life form in SIVA-T05 (Figure 3). The VLI consists of a virtual genome and functional automata. The virtual genome is a VP of the constitutive information group and corresponds to instruction

tape I in Figure 3, whereas the functional automata are VPs belonging to the functional module group and correspond to automata A, B, C, and FZ in Figure 3. The virtual genome encompasses the functions of preservation, replication, and transcription of structural and functional information about a VLI, while the functional automaton encompasses various life activities of the VLI, such as synthesis, decomposition, and reproduction.

The virtual genome consists of a sequence of four kinds of VM (W, X, Y, Z in Table 1) corresponding to the nucleotide in terrestrial life (Figure 3). In the virtual genome, five VMs constitute a functional unit, which serves as a Vcodon. Namely, each Vcodon is defined as corresponding to one of 18 kinds of VM (I, J, K, L; O, P, Q, R; 0–9 in Table 1) of the functional module group (i.e., virtual amino acid: VAA). The sequence of Vcodons defines the sequence of the VAAs in a functional automaton. The sequence information regarding all automata is described in the virtual genome. For the reproduction of a VLI, automaton B replicates the whole virtual genome, and automaton A synthesizes a functional automaton. Mutation can occur in either of these processes.

SIVA-T05 executes the functions of the automata described by the SIVA language as an interpreter by which life activities of VLIs are expressed. First, a functional unit consisting of a sequence of five VAAs serves as a <word> in the SIVA language. A <word> can be categorized as a functional word, which serves as an executable <command>, or as a temporary information word (Table 1). A <command> as a functional word covers a substantial part of the life activities of a VLI. One or more words constitute a <sentence>, which has to include zero or more <command>s and one <period> at the end. Before a <command>, a <sentence> can include one or more conditional phrases. When there is no conditional phrase in the <sentence>, <command>s are directly executed in the order described in the <sentence>. If a <sentence> includes any conditional phrases, a <command> is executed only when

Table1: Hierarchization of virtual biomolecules composing virtual life based on the complexity of the inter-atomic network.

Class name	Functional module group		Constitutive information group
Virtual biological polymer (VP)	Polymerized functional units		
<i>Functional unit</i>	Functional word (command)	Temporary information word (variable, relational operator etc.)	Virtual codon
Virtual biological monomer (VM)	O P Q R(4 kinds)	I J K L 0 1 2 3 4 5 6 7 8 9 (14 kinds)	W X Y Z(4 kinds)
Virtual organic biomaterial (VO)	A B C D (4 kinds/upper-case letter)		
Virtual inorganic biomaterial (VI)	a b c d (4 kinds/lower-case letter)		

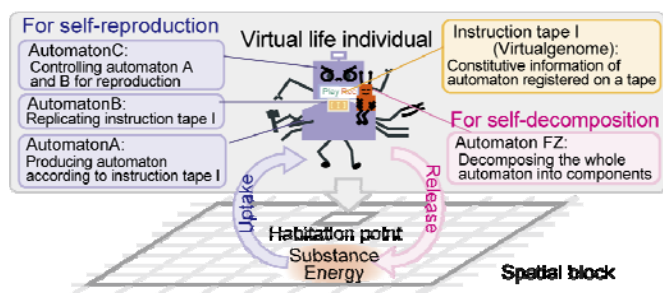


Figure 3. Relationship between life activities of virtual life individuals (VLIs) and the environment in SIVA-T05. Oohashi's SRSD automaton is implemented in the VLI in SIVA-T05. Each VLI consists of functional automata for self-reproduction [D (=A+B+C)], those for self-decomposition [FZ], and an instruction tape [ID+FZ] (i.e., a virtual genome) that is a blueprint of all the automata. Automaton A produces all the functional automata described in the virtual genome. Automaton B replicates the virtual genome. Automaton C constitutes a daughter VLI, combining the automata newly synthesized by automaton A and the virtual genome replicated by automaton B, and divides it from the parental VLI. Automaton FZ decomposes a VLI when the VLI encounters environmental conditions unsuitable for survival or when it lives out its life span. A VLI can reproduce itself by the uptake of substances and energy existing in the spatial block to which its habitation point belongs. During self-decomposition, the substances and the energy generated by the decomposition of virtual biomolecules constituting the VLI are restored to the spatial block. The occupied space is also released for utilization by another VLI.

all the conditional phrases are true but not when any of the conditional phrases is false. On the basis of these rules, a VLI can be programmed to undergo individual division when all conditions are satisfied, and to decompose itself when unfitness for its environment exceeds threshold level, etc.

Each VLI expresses its life activities by executing all <sentence>s during one time count (TC), the unit of virtual time in SIVA-T05. The order in which a VLI in the virtual ecosystem expresses its life activities within one TC is randomly determined at every TC. It takes at least 5 TCs for a newborn individual to reproduce itself in our current simulation experiments. Therefore, we use <passage duration> as a virtual time unit, which corresponds to the value of TC divided by 5.

When a VLI reproduces itself, it chooses a habitation point for a newborn VLI adjacent to its own habitation point. If the life activities of a newborn VLI fit the environmental conditions in the habitation point, it can also reproduce itself. If such activities do not do so, the newborn VLI decomposes itself prior to reproduction. Since certain mutations may accumulate as generation changes recur, certain offspring may emerge whose life activities fit environmental conditions differing slightly from those existing for their parents. Consequently, VLIs increase or decrease the size of their habitation point. (Oohashi et. al., 2009)

### 3) Experimental conditions

First, we designed a VLI of a mortal organism with a genetic program for death. This VLI has Automaton A, B, C and FZ as described in Figure 1 and 3, an initialization Automaton that produces the initial setting of the VLI, and a virtual genome corresponding to these Automata. On the basis of the PSD model (see Figure 1), the Automaton FZ, the mechanism for death, was designed to be activated when either of the following conditions is true: (1) unconformity between the VLI and its habitation environment or (2) the end of the life span of the VLI. We took advantage of this mechanism to design a VLI of an immortal organism, of which the value of both the conditional phrases of SIVA language for Automaton FZ were kept unchangeable at a false value and accordingly the functional words in SIVA language for self-decomposition in the FZ automaton were kept unchangeable at an inactivated state. If a mutation occurs in one of these conditional phrases and the value of either conditional phrase becomes changeable, it means that a mortal VLI is evolutionarily born. The functional words in SIVA language for self-decomposition of the mutant VLI will become activated, and the VLI will decompose itself when the above conditions become satisfied during the life of the VLI.

We seeded a single VLI that possessed this precursor of a genetic program for death in the center habitation point of the ecosystem with suitable environmental conditions and then conducted simulations of reproduction and evolution.

In the present simulation experiments, mutation of virtual genomes randomly occurs at the probability predetermined as a mutation rate. We investigated three mutation rates as follows: 0.005, 0.002 and 0.001. Mutation rates of the existing terrestrial lives are distributed from 10<sup>-4</sup> to 10<sup>-10</sup>. There is a tendency for a living organism with a small genome to exhibit a large mutation rate. For example, an organism with a genome of 104 molecules has a 10<sup>-4</sup> mutation rate. Virtual genomes of the VLIs in the present simulation experiments consist of 1275 molecules of VM, so we think the above configured mutation rates are within an appropriate range.

Consequently, we conducted 200, 500 and 800 simulations at mutation rates of 0.005, 0.002 and 0.001, respectively. The simulations were of 800 passage durations. Changes in size of the habitation area, number of individuals, and frequency of mutation were observed.

## Results

The rates at which mortal organisms evolutionarily emerged and survived are shown in Table 2. The denominators are the number of simulation trials including many cases in which no valid mutation occurred or no VLI of a mortal organism emerged within the 800 passage durations. The rates are 3.5%, 1.4%, and 0.25% for mutation rates of 0.005, 0.002, and 0.001, respectively. That is to say, when the genetic program for death was evolutionarily acquired, the individual possessing the program and its offspring did not always become extinct and survived within a certain probability.

When a VLI of a mortal organism survived, it and its offspring surpassed VLIs of an immortal organism and became prosperous without exception. Figure 4 shows successive changes of VLI distribution, number of



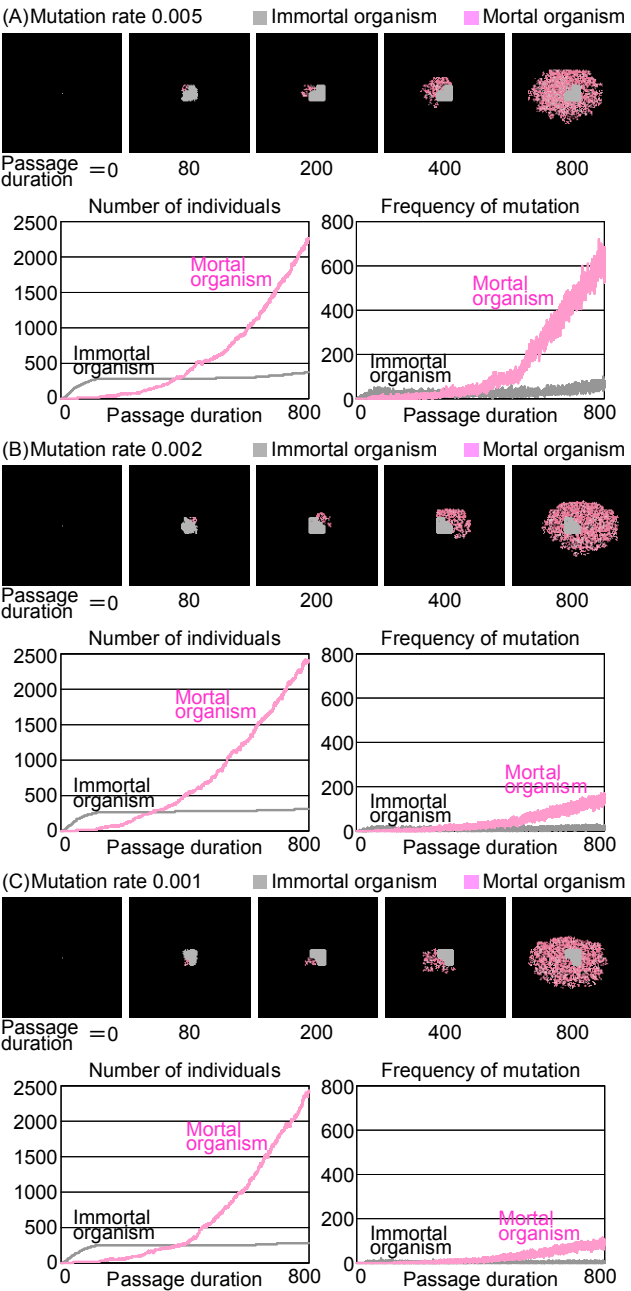


Figure 4. Evolutionarily emerging and surviving VLI of mortal organism certainly surpassed VLIs of immortal organism and became prosperous with adaptive divergence under various environmental conditions. Successive changes of individual distribution, the number of individual, and the frequency of mutation were illustrated. (A) 0.005 of mutation rate. (B) 0.002 of mutation rate. (C) 0.001 of mutation rate.

individuals, and frequency of mutation for each mutation rate. For example, for the mutation rate of 0.005 [Figure 4 (A)], a VLI with a genetic program for death emerged at the 30 passage durations' mark and produced offspring without extinction. In the case of 0.002 and 0.001 mutation rates

[Figure 4 (B), (C)], a VLI with a genetic program for death emerged at 11 and 29 passage durations respectively. Both produced offspring without extinction. Successive changes in the number of individuals and the frequency of mutation shown in Figure 4 demonstrate massive activities of mortal organisms compared to those of immortal organisms. The number of VLIs of a mortal organism grew at a sluggish pace shortly after emergence. However the mortal organisms extended their habitation area by degree, moved ahead of immortal organisms around the 300 or 400 passage duration mark, and then continued to extend their habitation area. There was no difference observed in the number of VLIs of a mortal organism introduced by the difference in mutation rate. We think the difference in the frequency of mutation of mortal organisms is reasonable because it may be introduced by the difference in mutation rates.

Table2: Probability of evolutionary emergence and survival of mortal organism

Mutation rate	Evolutionary emergence and survival	
	Frequency	Probability
0.005	7 times per 200 trial	3.5%
0.002	7 times per 500 trial	1.4%
0.001	2 times per 800 trial	0.25%

Discussion

1) Mortal organism survived and prospered within a certain probability

We carried out an evolutionary simulation experiment using our artificial ecosystem SIVA-T05, modeled for a finite, heterogeneous terrestrial environment and arranged in a biomolecular hierarchy. In many cases, we observed that when a mortal organism endowed with an evolutionarily acquired genetic program for death was born in a place in which immortal organisms already existed, the mortal organism, instead reproducing, became extinct by means of self-decomposition, overwhelmed by the indigenous immortal organisms. Nonetheless, our simulation process also demonstrated that some mortal organisms were evolutionarily appeared and managed to survive at a probability of 0.25% to 3.5% in accordance with mutation rates (Table 2). Furthermore, without exception, the mortal organisms that could overcome extinction thereafter prospered to the extent that they surpassed immortal organisms and continued to prosper, thanks to adaptive divergence under various environmental conditions. Although the probability of the survival and prosperity of the mortal organisms as shown in our simulations was low, it was, nonetheless, significant. Thus we can expect that mortal organisms might evolutionarily emerge, survive and prosper with adaptive divergence in other ecosystems under various environmental conditions while various ecosystems would repeatedly receive not a few opportunities for mutation. Considering the result of the experiment that a 0.25-to-3.5% probability for simulated ecosystems in which mortal

organisms prosper within a short duration of 800 passage durations applied to the terrestrial ecosystem, we believe that scale and heterogeneity of the earth's environment and length of time having elapsed during the evolution of terrestrial life and its concomitant ecosystem constitute sufficient probability for the possibility that mortal organisms could be evolutionarily selected and prosper terrestrially. Hence no inconsistency exists between our results and the experimental results described in our previous report (Oohashi et al. 2001).

## 2) Explanation of the superiority of mortal organisms

The transition of a number of individual organisms (Figure 4) indicates that the number of mortal organisms surpasses that of immortal organisms at the point in time after which 300-to 400 passage durations has elapsed, and that mortal organisms continue to prosper thereafter. How do mortal organisms overwhelm immortal organisms in this process? One interpretation of this phenomenon is as follows:

Immortal organisms dominate space and materials once they have been secured while the volume of resources to sustain life activities monotonically decreases. With less chance of reproduction in association with decrease of resources, chances for mutation as well as those for evolutionary adaptation are likewise reduced without limit.

On the other hand, mortal organisms release space for other organisms and return optimum parts for them to reutilize through self-decomposition upon termination of their mortal life. By doing so, equally benign or enhanced habitat environmental conditions can thus be secured for the all organisms including their own offspring in the ecosystem, which, in turn, will repeat the alternation of generation by utilizing finite space and materials. It is conceivable that due to accumulated mutations through the alternation of generations, new organisms emerge as a result of accelerated evolutionary adaptation in neighboring areas under environmental conditions that had not previously permitted the existence of earlier generations.

Independent of the studies that we have undertaken since 1987 (Oohashi et al., 1987, 1996, 1999, 2001, 2009, 2011), Todd implemented artificial death in his ALife system (Todd, 1993, 1994), and those experiments supported the recognition shared with us that death affords another entity its space in which to exist, and that death, accordingly, is essential throughout the ongoing evolutionary process. Nevertheless, the model of death constructed by Todd differs from our model of death in two patently obvious respects. First, death in Todd's model affords no process by which the organism might decompose itself into constituent parts for the efficient and collective reutilization of other organisms, which is an essential feature of our model. Second, the death of an individual in Todd's model appears as a probabilistic phenomenon, or as a given result controlled by the simulation system, in sharp contrast to the activation of death in our model, which is a process genetically regulated in the individual that starts from detection either of the end of its life span or of excess unconformity with the environment. Consequently, it would be difficult to use the ALife system as constructed by Todd to investigate the evolutionary emergence of death itself.

It is noteworthy that the mechanism of programmed self-decomposition, observed as being evolutionarily selected in this study, accords benefits not only to direct offspring but

also to all organisms of the entire ecosystem. It is difficult to produce a tenable explanation for this phenomenon based only on the "selfish gene" paradigm.

Programmed self-decomposition has been observed as a life phenomenon of existent terrestrial life as previously reported (Oohashi et al, 1987, 2009). The gradual consolidation of these complementary approaches—ALife simulations and biological experiments—will likely throw added light on this topic in the future.

## 3) Conclusion

The evolutionary simulations using our artificial ecosystem SIVA-T05 show that, if mortal organisms evolutionarily acquire a genetic program for autonomous death and then appear among a population of immortal organisms, such mortal organisms, endowed as they are with a genetic program for autonomous death, can survive and will surpass immortal organisms lacking autonomous death and will prosper with adaptive divergence under various environmental conditions within a certain probability.

The above results thus support our hypothesis that originally immortal organisms evolve into mortal organisms by acquiring a new genetic program for autonomous death.

## References

- Odum, E. P. (1971). *Fundamentals of Ecology* (3rd ed.). Philadelphia, PA: W.B. Saunders Company.
- Oohashi, T., Nakata, D., Kikuta, T., and Murakami, K. (1987). Programmed self-decomposition model. *Kagakuikisoron (in Japanese)*, 18(2): 21-29.
- Oohashi T., Maekawa T., Ueno O., and Honda M. (2011). The supremacy of the altruistic gene: Terrestrial life has succeeded in breaking through evolutionary deadlock. *Kagaku (in Japanese)*, 81(1):83-90.
- Oohashi, T., Maekawa, T., Ueno, O., Kawai, N., Nishina, E., and Shimohara, K. (2001). Artificial life based on the programmed self-decomposition model: SIVA. *Journal of Artificial Life and Robotics*, 5(2):77-87.
- Oohashi, T., Maekawa, T., Ueno, O., Nishina, E., Kawai, N. (1999). Requirements for immortal ALife to exterminate mortal ALife in one finite, heterogeneous ecosystem, *Proceedings of the 5th European Conference on Artificial Life*, pages 49-53. Springer-Verlag, London, UK.
- Oohashi, T., Sayama, H., Ueno, O., and Maekawa, T. (1996). Artificial life based on programmed self-decomposition model. *ATR Technical Report*, TR-H-198.
- Oohashi T., Ueno O., Maekawa T., Kawai N., Nishina E., and Honda M.(2009). An Effective hierarchical model for the biomolecular covalent bond: An approach integrating artificial chemistry and an actual terrestrial life system. *Artificial Chemistry Special Issue of Artificial Life*, 15(1):29-58.
- Suzuki, H. (2004). Network artificial chemistry - Molecular interaction represented by a graph. In M. Bedau, P. Husbands, T. Hutton, S. Kumar & H. Suzuki (Eds.), *Proceedings of Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9) Workshop and Tutorial* (pp. 63-70), Boston, MA.
- Todd, P. M. (1993). Artificial Death. *Second European Conference on Artificial Life (ECAL93)*, pages 1048-1059.
- Todd, P. M. (1994). Artificial Death. In C. Schneider (Ed.), *Jahresring 41*, pages 90-107. Verlag Silke Schreiber, Munich.
- Von Neumann, J. (1951). The general and logical theory of automata. In L. A. Jeffress (Ed.), *Cerebral mechanisms in behavior - The Hixon symposium*. pages. 1-41, John Wiley & Sons, New York.



# Evolution of Partner Selection

Pedro Mariano<sup>1</sup> and Luís Correia<sup>1</sup>

<sup>1</sup>LabMAG – Dep. de Informática, Faculdade de Ciências, Universidade de Lisboa, Portugal  
plmariano@di.fc.ul.pt    Luis.Correia@di.fc.ul.pt

## Abstract

Partner selection is a mechanism that promotes sustainability of cooperators in cooperative dilemmas. In this paper we investigate the conditions that favour the evolution of a particular partner selection model that can be applied to any  $n$ -player game. The model allows a player to select partner combinations that satisfy his preferences. A limit case of the model is random choice of partners. Model parameters are under evolutionary control. We present simulations of our model that show evidence of the evolution of partner selection instead of random choice.

## Introduction

In social interactions one of the main sources of distress is the proliferation of non-cooperative elements. A small percentage of unsocial behaviour is well accepted or even beneficial (Semmann et al., 2003). However, an unlimited growth of the percentage of free-riders is detrimental to cooperation and therefore to the maintenance of a society as a whole.

The study of social interactions has been modelled by several games presenting social dilemmas. For instance we have Iterated Prisoner's Dilemma (IPD), Ultimatum, Investment, Centipede, Public Good Provision (PGP) and Give-Take (Gintis, 2000; Fudenberg and Tirole, 1991; Axelrod, 1997; Mariano and Correia, 2002). Theoretical analysis of these games predicts the prevalence of exploiters or non-social behaviour in general (Gintis, 2000).

Several approaches have been developed in order to limit proliferation of free-riders. Some of them use game specific strategies while others fall into mechanism design. In the former category, we have tit-for-tat as an example of a strategy to play IPD that in a variety of conditions is able to resist non-cooperative players. In the latter we have the possibility of partner selection (Izquierdo et al., 2010; Santos et al., 2006; Aktipis, 2004).

In this paper we investigate the conditions that favour the evolution of partner selection in any symmetrical  $n$ -player game. In particular, we examine the model proposed in Mariano and Correia (2010). That model assigns probabilities to combinations of partners that are updated in a process similar to Hebbian learning. The process motivation is that, in the

long run, cooperative players mostly select partners among themselves. When a positive interaction occurs, instead of reinforcing probabilities of combinations, probabilities remain unchanged. When a negative interaction occurs, the combination is replaced and its probability is decreased. As a result probabilities of combinations with positive interactions absorb decreasing probabilities. By positive interaction we mean that a player considers the result as acceptable or the interaction as cooperative. The model can be applied to any  $n$ -player game with any type of strategy (deterministic or stochastic).

## Related Work

It has been reported in human experiments (Barclay and Willer, 2007; Coricelli et al., 2004; Ehrhart and Keser, 1999) that if players are able to select their partners they will seek cooperative partners while escaping free riders. In Price (2006) the author refers that in experiments involving human subjects, people tend to cooperate more when they can choose their interaction partners and, in that case, they cooperate when they perceive altruistic behaviour.

There is research on partner selection (Izquierdo et al., 2010; Pacheco et al., 2006; Santos et al., 2006; Zimmermann and Eguíluz, 2005; Aktipis, 2004; Semmann et al., 2003; Hauert et al., 2002; Stanley et al., 1995; Orbell and Dawes, 1993; Vanberg and Congleton, 1992) but this characteristic is granted in the model, i.e. players cannot choose between random partner allocation (Suzuki and Akiyama, 2008; Axelrod and Hamilton, 1981) or having the possibility to select with whom they will play. Moreover, these models are often tailored for a specific game such as PGP or IPD (Izquierdo et al., 2010; Aktipis, 2004).

Research similar to ours is Santos et al. (2006) and Pacheco et al. (2006) where population structure is able to evolve. Players are embedded in a network. If a player can change his links, selection favours cooperators that prefer to maintain links with their kin and to drop links with defectors. However, their findings were done in 2-player games and they only considered two types of strategies.

## Model Description

The model of partner selection presented in Mariano and Correia (2010) is characterised by two vectors. One,  $\mathbf{p}$ , contains combinations of  $n - 1$  partners drawn from a set of candidate partners, which constitute the player's neighbourhood  $\mathcal{N}$ . Each combination is assigned a probability stored in vector,  $\mathbf{c}$ . In that paper three update policies of the above vectors are compared. In the present paper, we use the policy that has given the best results. In this policy, after a player plays a game with a combination drawn from vector  $\mathbf{c}$  it compares the utility obtained  $u$  with parameter  $u_T$  and updates vector  $\mathbf{p}$ . The probability of the selected combination,  $k$ , is updated as follows:

$$p_k^{t+1} = \begin{cases} \delta p_k^t & \text{if } u < u_T \\ p_k^t & \text{if } u \geq u_T \end{cases} . \quad (1)$$

The probabilities of other combinations are updated as follows:

$$p_i^{t+1} = \begin{cases} p_i^t + \frac{(1 - \delta)p_k^t}{l - 1} & \text{if } u < u_T \\ p_i^t & \text{if } u \geq u_T \end{cases} , \quad (2)$$

where  $i \neq k$ , in order to maintain sum to unit and  $\delta$  represents the probability decrease factor.

If the utility is lower than threshold  $u_T$ , slot  $k$  of vector  $\mathbf{c}$  is replaced by a randomly generated combination, different from the ones in the other slots. Players of the new combination are randomly selected from  $\mathcal{N}$ .

Both vectors  $\mathbf{c}$  and  $\mathbf{p}$  have the same length represented by parameter  $l$ . This model has two particular cases of partner selection. When  $l = 0$  the player randomly picks  $n - 1$  partners from  $\mathcal{N}$  to play a game. The specific case of  $l = 1$  is similar to the model presented in Aktipis (2004) and Izquierdo et al. (2010). In these works, which only consider IPD (a game with two players) if a player is not happy, he moves away seeking a new partner. In our case, a new random combination of partners is selected.

## Player Chromosome

The description of the model has shown that it can handle random partner allocation as well as selection of best partner combinations. As we are using an evolutionary algorithm, the model parameters, namely  $l$ ,  $\delta$  and  $u_T$ , are part of the player's chromosome. In our simulations the domain of  $l$  is  $\{0, 1, \dots, \bar{l}\}$ , where  $\bar{l}$  represents the maximum value of  $l$  and the domain of  $\delta$  is  $[0, 1]$ . The update policy is based on private information, namely the utility the player assigns to a specific partner combination. In this paper we simplify and assume  $u = \pi$ , the utility is equal to the payoff  $\pi$  ascribed by the specific game used. Therefore the domain of  $u_T$  is  $[\underline{\pi}, \bar{\pi}]$ , where  $\underline{\pi}$  and  $\bar{\pi}$  are, respectively, the lowest and highest payoff of the game. Besides these three parameters, the chromosome also contains the strategy,  $s$ , used to play the

$s$	strategy
$l$	size of vectors $\mathbf{p}$ and $\mathbf{c}$
$u_T$	utility threshold
$\delta$	probability decrease factor

Table 1: Player's chromosome

game. When talking about the chromosome we may designate the coded parameters as variables or genes. Table 1 summarises player's chromosome.

## Evolutionary Setup

A plain genetic algorithm (Holland, 1975) with players' fitness as the total payoff obtained by a player favours players that are selected more often, typically cooperators. On the other hand, if we average players' payoffs other types of players are favoured. For instance, an exploiter that played a single game and obtained the highest payoff is favoured compared to cooperative players that played more games among themselves, which produces a lower average.

Artificial Life systems such as AVida (Misevic et al., 2006), Tierra (Ray, 1992) or Polyworld (Yaeger, 1994) do not have an explicit fitness function. These systems are considered when the goal is the simulation of open-ended evolution (Chaumont and Adami, 2010). Individuals must continuously adapt their strategy to the environment they are faced with. Typically, individuals must manage their energy in order to survive and pass their genes to their offspring.

Here we use a similar model but we frame it in the context of game theory. We consider that players obtain energy by playing some game  $\mathcal{G}$ . A player reproduces when his energy reaches some threshold. Every newborn player starts with zero energy. A player's energy is incremented by the payoff  $\pi$  he obtains. In order to avoid negative energies due to negative payoffs, we adjust the payoff by the lowest payoff obtained in the game,  $\underline{\pi}$ . Summing up, the energy,  $e$ , of a player is updated as:

$$e^{t+1} = e^t + \pi - \underline{\pi} . \quad (3)$$

Whenever a player's energy reaches the reproduction threshold,  $e_R$ , he produces an offspring. Reproduction is asexual and the offspring is a clone of the parent subject to mutation. The parent's energy goes back to zero.

The mutation operator is similar for all genes. Parameter  $l$  is perturbed by a discretized normal distribution with mean zero and standard deviation  $\bar{l}/2$ . Parameter  $u_T$  is modified by a normal distribution with mean zero and standard deviation  $(\bar{\pi} - \underline{\pi})/2$ . Parameter  $\delta$  is perturbed by a normal distribution with zero mean and standard deviation 0.5.

Summarising, a player's phenotype is characterised by his strategy, the probability and combination vectors, his neighbourhood and his energy. We also record a player's age. Table 2 shows these parameters. When a player is born, vector

$a$	age
$s$	strategy
$\mathbf{p}$	probability vector
$\mathbf{c}$	combination vector
$\mathcal{N}$	neighbourhood
$e$	energy

Table 2: Player’s phenotype

$\mathbf{c}$  is initialised with  $l$  random combinations of partners and vector  $\mathbf{p}$  is initialised with constant value  $l^{-1}$ .

## Environment

There are different artificial environments that influence how players interact. Research on cooperation uses toroidal lattices (Nowak et al., 2004), well-mixed populations (Pacheco et al., 2006), or small-world networks. Population structure influences the evolution and stability of cooperation. We opted for a well-mixed population, which means that a player can draw a combination from all the other players. Formally, for every player  $\alpha$  in population  $\mathcal{P}$  we have  $\{\alpha\} \cup \mathcal{N}_\alpha = \mathcal{P}$ . This is a typical structure in small communities (Price, 2006).

A simulation is composed of several rounds,  $N_R$ . In each round, all players select a combination of partners from their combination vector  $\mathbf{c}$  using their probability vector  $\mathbf{p}$ . They play the game  $\mathcal{G}$ . For each played game, all participants update their energy as defined by equation (3). The player that selected the partners is the only one that updates his probability and combination vectors, according to equations (1) and (2). The other players may not know all their partners. Only the selecting player has all the players in his combination vector. This approach prevents players from copying others’ combinations vectors.

The next step in a round is reproduction. All players that have reached the reproduction threshold generate one offspring. These players have their energy reset to zero.

Since reproduction increments population size, we need a mechanism to avoid an infinite growth of players. In the end of each round, a player may die with a probability given by the following sigmoid function:

$$P(\text{player dies}) = \frac{1}{1 + e^{B - |\mathcal{P}| - a}} \quad (4)$$

where  $B$  represents the carrying capacity,  $|\mathcal{P}|$  is the population size and  $a$  is the player’s age. Not only a player dies from overcrowding, but also he dies from old age. This implies that set  $\mathcal{N}$  may vary from round to round with a strong dependency on  $B$ . Since we considered a well-mixed population,  $\mathcal{N}$  is virtually the size of the population. Parameters that describe the overall behaviour of a simulation are presented in table 3.

$\mathcal{G}$	$n$ -player game
$B$	carrying capacity
$e_R$	reproduction threshold
$\mathcal{P}^0$	initial population
$N_R$	number of rounds

Table 3: Simulation parameters

## Comments

Players that are only exploited and cannot find sufficient co-operators, will not be able to reproduce. Also a population composed of a majority of exploiters may go extinct if the reproduction threshold is high.

The ratio  $R_1 = e_R / (\bar{\pi} - \pi)$  represents the minimum number of games a player has to play to reproduce. The higher the former value, the more pronounced the effect of partner selection. It takes some time for the probability vector to converge to a situation where only cooperators are present in the best combinations of the combination vector. This was observed in a situation where set  $\mathcal{N}$  is static (Mariano and Correia, 2010). In this paper we show that this may also happen in dynamic populations, meaning with variable  $\mathcal{N}$ . Notice that a cooperator will reproduce increasingly faster until the convergence of the probability vector.

An Evolutionary Stable Strategy (ESS) of some game  $\mathcal{G}$  depends on the evolutionary mechanism. For instance, in a context of infinite populations where players play infinitely often (Hofbauer and Sigmund, 1998), defection is the ESS of the PGP game.

Using the energy model that we presented without partner selection (partners randomly picked) all players will play approximately the same number of games. If the game  $\mathcal{G}$  is symmetric, its Nash Equilibrium (NE) will be maintained with this energy model. If some player deviates from the NE, energy obtained per game diminishes and consequently he will take longer to reach the reproduction threshold. This means that he will produce less offspring compared to those that stick to the NE. Due to the carrying capacity, the deviating player and his offspring have more chances of disappearing. The bottom line, is that in PGP with our energy model defection is still the ESS. However, with partner selection this may change. If a player can choose his partners, there may exist other ESS, namely cooperation, in the PGP game. This results from cooperators selecting preferably among themselves.

We have used a well-mixed population. Even in this case, since players select their partners, they are effectively constructing a network of contacts. The minimum and maximum number of contacts a player may have depend on  $n$  and  $l$ . The combination vector  $\mathbf{c}$  can have  $l$  distinct combinations differing in a single partner, yielding a minimum value of  $n + l - 2$  contacts. On the other end, all players in every combination may be unique, yielding a maximum

$n$	3	4	5	6	7	
$B$	100	110	120	130	140	150
$e_R$	50	60	70	80	90	100

Table 4: Parameters values used in the simulations.

value of  $(n - 1)l$  contacts.

This contrasts with recent work that considered other types of population structure such as small-world and scale-free (Pacheco and Santos, 2005). With structure the contact limits above defined may be further reduced.

## Experimental Analysis

The capability of the model we present to evolve partner selection can be assessed by tracking parameters  $l$ ,  $u_T$  and  $\delta$ . On the other hand, sustainability of cooperation can be measured by counting the number of cooperators that appear in a simulation.

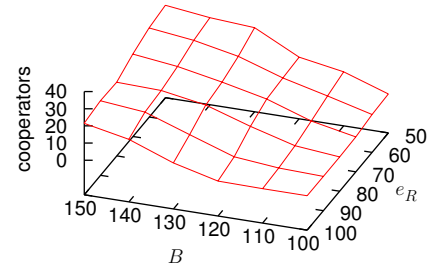
### Game

We have performed simulations using the PGP game (Boyd et al., 2003; Hauert et al., 2002). This game is commonly studied to analyse cooperative dilemmas. Moreover, it is a  $n$ -player game. It is considered a generalisation of the Prisoner's Dilemma (PD) game to  $n$  players. In the PGP game, a player that contributes to the good, incurs a cost  $c$ . The good is worth  $g$  for each player. Let  $x$  be the proportion of players that provide the good. The payoff of a player that provides the good is  $gx - c$  while players that defect get  $gx$ . The game has a single iteration. The strategy used by a player is probabilistic and is defined by probability  $s$  to provide the good. We assume that the utility of a player is equal to its payoff. In the simulations we set  $g = 1$  and  $c = 0.4$ . The number of players in a game varied between three and seven. In this game, defection is the Nash Equilibrium and it is also the unique ESS.

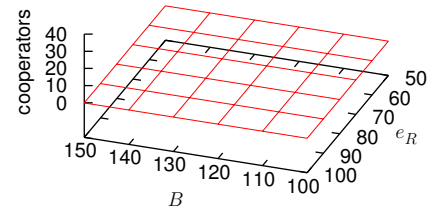
### Tested Parameters

We have varied the carrying capacity and some parameters that influence the number of games a player has to play in order to reproduce. The latter is directly influenced by the reproduction threshold  $e_R$  but also by the number of players per game,  $n$ . Table 4 shows the values of the tested parameters thus giving an overview of the conditions where the evolution of partner selection was tested.

The size of initial population is 20. Those players all have the same chromosome: ( $s = 1, l = 0, u_T = \pi, \delta = 0$ ), i.e. players are cooperative but perform random selection of partners. Whenever a new offspring is born, mutation occurs with probability 0.1. Mutation of genes  $l$ ,  $u_T$  and  $\delta$  has already been described. The maximum value of gene  $l$  was 10. Gene  $s$  is altered by a normal distribution with mean zero and standard deviation 0.1. Each simulation run



(a) 3-player PGP.



(b) 4-player PGP.

Figure 1: Average number of cooperators in the last round.  $e_R$  is the energy required for reproduction and  $B$  is the carrying capacity.

consists of  $N_R = 10^5$  rounds. For statistical purposes, each result was taken from 30 independent runs, except otherwise noted.

## Results

One major outcome was the identification of conditions for the survival of cooperators. This is important because in the plain PGP defection is the ESS. The use of partner selection modified this situation. Since we are using probabilistic strategies, we classified a strategy as cooperating if it cooperates more than 90% of the time. This is a strict threshold and results could improve if it was lower.

The survival of cooperators depends mostly on the number of players in a game. With 3-player PGP cooperators survive, but not with 4 or more players per game (see figure 1). It has been shown in Mariano and Correia (2010) that the number of possible combinations of partners grows exponentially with the number of players per game,  $n$ , and the number of candidate partners,  $\mathcal{N}$ , which is the size of the neighbourhood. Now, in a well-mixed population ( $\mathcal{N}$  is the size of the population) with  $n = 4$ , the difficulty to find a favourable combination is already too high for cooperators to survive. In general, if the set of candidate partners is big and the number of players in a game is high, there are more

$n$	$\underline{\pi}$	$\bar{\pi}$	$\bar{\pi} - \underline{\pi}$	#C= $n$ $\Delta e$ C	#C= $n-1$ $\Delta e$ C	#D=1 $\Delta e$ D
3	-.27	.67	.93	.67	.33	.93
4	-.35	.75	1.10	.75	.50	1.10
5	-.40	.80	1.20	.80	.60	1.20
6	-.43	.83	1.27	.83	.67	1.27
7	-.46	.86	1.31	.86	.71	1.31

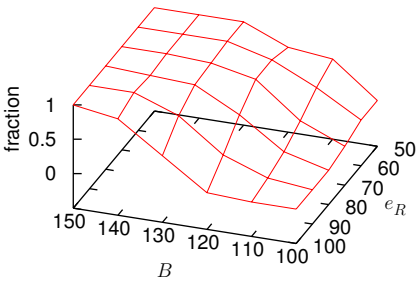
Table 5: Payoff range and energy obtained per number of players. The last three columns contain the energy obtained by a cooperator, represented by letter C, and by a defector, represented by letter D. In the first situation (#C=  $n$  column) all  $n$  players cooperate, while in the second (last two columns) all but one player cooperate.

combinations of players to explore. In these cases, the partner selection model requires more time to find the correct partner combination, which may not be available even with large life span and low reproduction threshold.

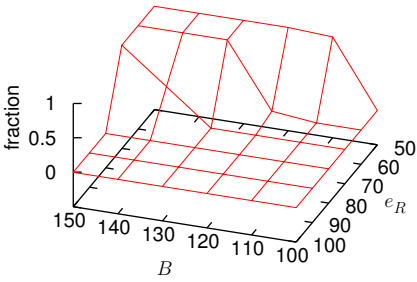
Results also show that there are conditions where the population decreases until there are not enough players to play a game. Such occurrence of extinctions depends on the number of players, energy required to reproduce and parameter  $B$ , as shown in figure 2. Extinctions increase with increasing  $e_R$  and decreasing  $B$ . While parameter  $B$  can be interpreted as a carrying capacity, it can also be interpreted as a player’s average life span (see equation (4)). With low  $B$  values, high  $e_R$  and small  $n$  it is improbable that a player can attain sufficient energy to reproduce during his life span. This situation leads to high extinction rates.

We have already seen that 3-player PGP is the only case where cooperators survive. When we go to 4-or-more-player PGP the only survivors are defectors. In 4-player PGP, cooperators are early on wiped out by exploiters and the remaining exploiters cannot obtain sufficient energy to reproduce and die of old age. However, with growing  $n$  the probability of extinctions diminishes. This is due to the fact that each player is chosen more often to play by his neighbours. Therefore he may be able to attain the reproduction threshold,  $e_R$ , even when parameter  $B$  is low. In a population composed of only defectors, the payoff obtained by each one is zero. However, even in this situation, due to how energy is calculated (see equation (3)), defectors gain some energy. The more players in a game the more energy defectors obtain. Table 5 shows minimum and maximum payoff values for the tested number of players.

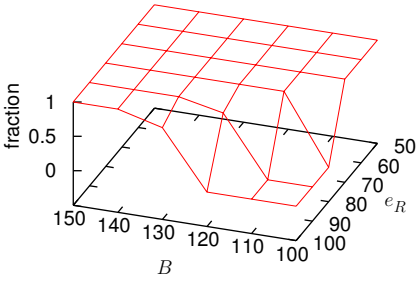
In the case of 3-player PGP, we analysed the evolution of the other three variables (genes) of the partner selection model, see figure 3. The pool size,  $l$ , increases from zero and stabilises around five. Variable  $\delta$  also increases from zero and stabilises around 0.5. As for  $u_T$  it increases, stabilising just under the Pareto payoff obtained by a cooperator playing with only cooperators. The fact that these pa-



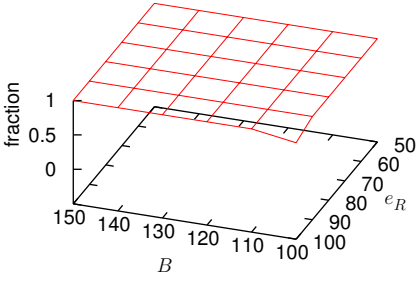
(a) 3-player PGP.



(b) 4-player PGP.



(c) 5-player PGP.



(d) 6-player PGP.

Figure 2: Percentage of simulations without extinctions.  $e_R$  is the energy required for reproduction and  $B$  is the carrying capacity.



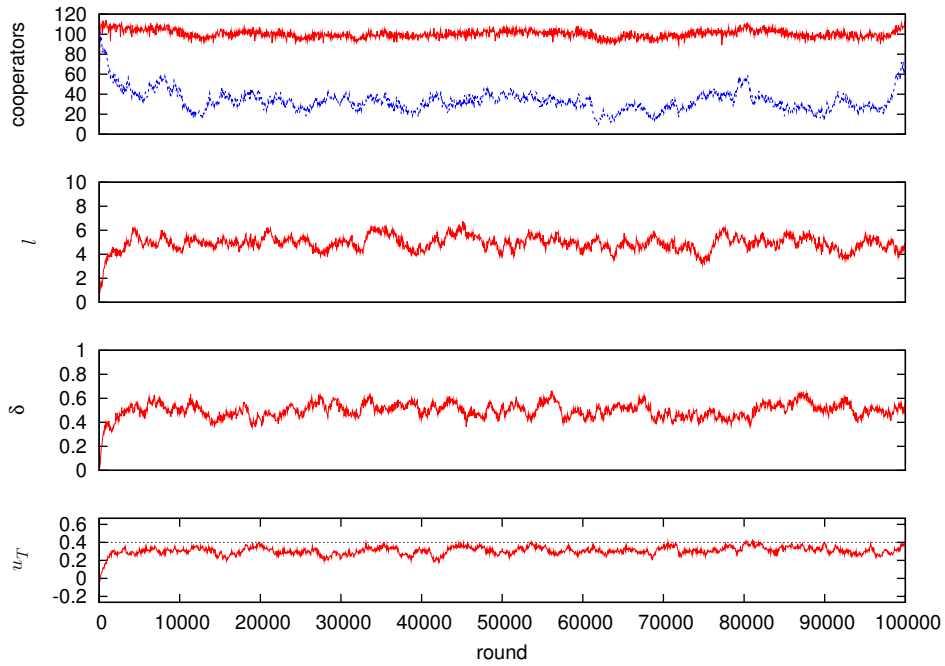


Figure 3: Example of a single simulation run (3-player PGP, carrying capacity  $B$  is 150, reproduction threshold  $e_R$  is 50) where cooperators are able to persist. In the cooperators plot, red solid line is population size and blue dashed line is number of cooperators (which fluctuates significantly without a corresponding influence on the average). In the plot of  $u_T$  the horizontal dashed line corresponds to the Pareto payoff. Results are plotted every 50 rounds.

rameters stabilise around some values means that there is no random drift. We confirmed such finding by measuring the variables under different  $e_R$  and  $B$  values, see figure 4. Remarkably, these variables are almost constant across all the values experimented for reproduction threshold,  $e_R$ , and  $B$ . Moreover, the memory length for partner combinations,  $l$ , is approximately 5, which is a quite small value. The fact these variables remain constant under different conditions reflects that a cooperator doing partner selection can find an adequate choice of partners, given time to achieve it.

## Conclusions

We have analysed the conditions for the evolution of a partner selection model. With such a model, the average number of games played by some player depends on his characteristic. Cooperators that select among themselves play more often compared to defectors. Reproduction was based on an energy model. Players reproduce when they attain some reproduction threshold. In order to contain the population under some limits, players die from overcrowding and old age.

The results show that cooperators are able to persist in a population even if with low percentages. These results were only possible due to partner selection. Cooperators persistence was only observed in some conditions, namely, 3-player PGP, high carrying capacity and small reproduction

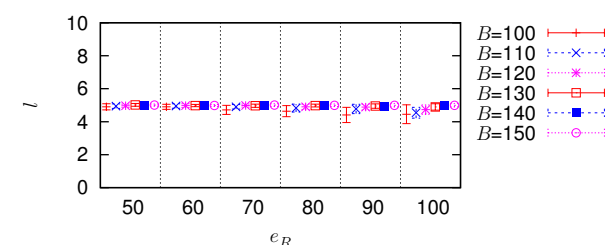
threshold. In other conditions, we observed that defection, which is the ESS, was the sole strategy present in the population. The evolutionary dynamics of partner selection did not show any random drift in its variables. In fact the model is quite robust since the memory of partner combinations,  $l$ , the probability decrease factor,  $\delta$ , and the utility threshold,  $u_T$ , are virtually independent of the carrying capacity,  $B$ , and the reproduction threshold,  $e_R$ .

We have shown the evolution of cooperators in the PGP. In contrast with others, (Izquierdo et al., 2010; Pacheco et al., 2006; Santos et al., 2006; Aktipis, 2004), this was obtained with stochastic strategies. Previous work has focused in specific games with only two strategies.

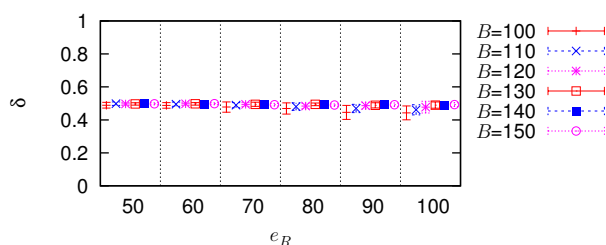
We are currently investigating how cooperators can persist in games with more than three players. Preliminary results show that with increasing carrying capacity cooperators live longer. Another possibility is to decouple the chance of player survival in two events: one for overcrowding and another for old age.

The fact that players are able to select with whom they play means that this model is suitable to study the emergence of niches. Suppose a game has multiple strategies to cooperate. This model of partner selection may favour the emergence of groups of players, where each group uses one of the different cooperating strategies available.

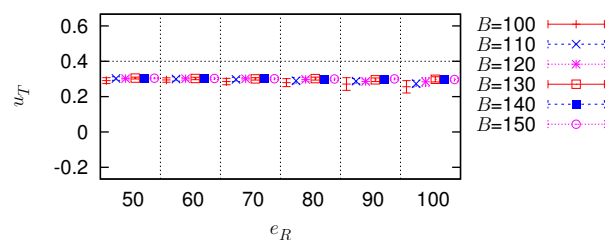
We have used a well-mixed population. However, this



(a) Average pool size.



(b) Average  $\delta$ .



(c) Average  $u_T$ .

Figure 4: Plots show average data per simulation round across reproduction threshold (drawn on horizontal axis) and parameter  $B$  (each value has a specific point). Data are taken from simulations with 3-player PGP.

does not preclude the appearance of a network of players. With partner selection, a player is restricted to interact only with the partners in his combination vector. The use of another population structure, such as small-world, for instance, reduces the number of players available to form partner combinations. Consequently the number of partner combinations will be more limited. If we take into consideration the results with 3-player PGP partner selection may not evolve in some situations. This occurs specially for small populations (small  $B$ ) and high reproduction energy,  $e_R$ . However, this is one avenue for future work.

## References

- Aktipis, C. A. (2004). Know when to walk away: contingent movement and the evolution of cooperation. *Journal of Theoretical Biology*, 231:249–260.
- Axelrod, R., editor (1997). *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*. Princeton Studies in Complexity. Princeton University Press.
- Axelrod, R. and Hamilton, W. D. (1981). The evolution of cooperation. *Science*, 211:1390–1396.
- Barclay, P. and Willer, R. (2007). Partner choice creates competitive altruism in humans. *Proceedings of the Royal Society B*, 274:749–753.
- Boyd, R., Gintis, H., Bowles, S., and Richerson, P. J. (2003). The evolution of altruistic punishment. *Proceedings of the National Academy of Sciences*, 100(6):3531–3535.
- Chaumont, N. and Adami, C. (2010). Potential and promise of open-ended evolution in artificial life. In Fellermann et al. (2010), pages 429–430.
- Coricelli, G., Fehr, D., and Fellner, G. (2004). Partner selection in public goods experiments. *Journal of Conflict Resolution*, 48(3):356–378.
- Ehrhart, K.-M. and Keser, C. (1999). Mobility and cooperation: On the run. CIRANO Working Papers 99s-24, CIRANO.
- Fellermann, H., Dörr, M., Hanczyc, M. M., Laursen, L. L., Maurer, S., Merkle, D., Monnard, P.-A., Stoy, K., and Rasmussen, S., editors (2010). *Artificial Life XII: Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*. MIT Press.
- Fudenberg, D. and Tirole, J. (1991). *Game Theory*. MIT Press.
- Gintis, H. (2000). Strong reciprocity and human sociality. *Journal of Theoretical Biology*, 206:169–179.
- Hauert, C., Monte, S. D., Hofbauer, J., and Sigmund, K. (2002). Volunteering as red queen mechanism for cooperation in public goods games. *Science*, 296:1129–1132.
- Hofbauer, J. and Sigmund, K. (1998). *Evolutionary Games and Population Dynamics*. Cambridge University Press.
- Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan.



- Izquierdo, S. S., Izquierdo, L. R., and Vega-Redondo, F. (2010). The option to leave: Conditional dissociation in the evolution of cooperation. *Journal of Theoretical Biology*, 267(1):76–84.
- Mariano, P. and Correia, L. (2002). The effect of agreements in a game with multiple strategies for cooperation. In Standish, R., Bedau, M. A., and Abbass, H. A., editors, *Artificial Life VIII: Proceedings of the Eighth International Conference on Artificial Life*, pages 375–378. MIT Press.
- Mariano, P. and Correia, L. (2010). Partner selection: Finding the right combination of players. In Fellermann et al. (2010), pages 852–859.
- Misevic, D., Ofria, C., and Lenski, R. E. (2006). Sexual reproduction reshapes the genetic architecture of digital organisms. *Proceedings of the Royal Society: Biological Sciences*, 273(1585):457–464.
- Nowak, M. A., Sasaki, A., Taylor, C., and Fudenberg, D. (2004). Emergence of cooperation and evolutionary stability in finite populations. *Nature*, 428:646–650.
- Orbell, J. M. and Dawes, R. M. (1993). Social welfare, cooperators’ advantage, and the option of not playing the game. *American Sociological Review*, 58(6):787–800.
- Pacheco, J. M. and Santos, F. C. (2005). Network dependence of the dilemmas of cooperation. In Mendes, J. F. F., editor, *Science of Complex Networks: From Biology to the Internet and WWW: CNET 2004*, volume 776, pages 90–100. AIP Conference Proceedings.
- Pacheco, J. M., Traulsen, A., and Nowak, M. A. (2006). Active linking in evolutionary games. *Journal of Theoretical Biology*, 243:437–443.
- Price, M. (2006). Monitoring, reputation, and “greenbeard” reciprocity in a Shuar work team. *Journal of Organizational Behavior*, 27:201–219.
- Ray, T. S. (1992). An approach to the synthesis of life. In Langton, C. G., Taylor, C., Doyne, J. D. F. J., and Rasmussen, S., editors, *Artificial Life II: Proceedings of the Second Conference on Artificial Life*, pages 371–408. Addison-Wesley.
- Santos, F. C., Pacheco, J. M., and Lenaerts, T. (2006). Cooperation prevails when individuals adjust their social ties. *PLoS Comput Biol*, 2(10):e140.
- Semmann, D., Krambeck, H.-J., and Milinski, M. (2003). Volunteering leads to rock-paper-scissors dynamics in a public goods game. *Nature*, 425:390–393.
- Stanley, E. A., Ashlock, D., and Smucker, M. D. (1995). Iterated prisoner’s dilemma with choice and refusal of partners: Evolutionary results. In Morán, F., Moreno, A., Merelo, J. J., and Chacón, P., editors, *Advances in Artificial Life*, volume 929 of *Lecture Notes in Computer Science*, pages 490–502. Springer-Verlag.
- Suzuki, S. and Akiyama, E. (2008). Chaos, oscillation and the evolution of indirect reciprocity in n-person games. *Journal of Theoretical Biology*, 252:686–693.
- Vanberg, V. J. and Congleton, R. D. (1992). Rationality, morality, and exit. *The American Political Science Review*, 86(2):418–431.
- Yaeger, L. (1994). Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or PolyWorld: Life in a new context. In Langton, C. G., editor, *Proceedings of the Workshop on Artificial Life (ALIFE ’92)*, volume 17 of *Sante Fe Institute Studies in the Sciences of Complexity*, pages 263–298, Reading, MA, USA. Addison-Wesley.
- Zimmermann, M. G. and Eguíluz, V. M. (2005). Cooperation, social networks, and the emergence of leadership in a prisoner’s dilemma with adaptive local interactions. *Phys. Rev. E*, 72(5):056118.

# Evolutionary attributes of simulated prebiotic metabolic networks

Omer Markovitch<sup>1</sup> and Doron Lancet<sup>1</sup>

<sup>1</sup>Department of Molecular Genetics, Weizmann Institute of Science, Rehovot 76100, Israel  
omermar@weizmann.ac.il

## Abstract

A metabolism-first scenario for the origin of life entails that as early as replicating entities have emerged prebiotically, they must have constituted relatively complex molecular networks, arising via spontaneous accretion of assemblies of simpler organic molecules. While it is widely accepted that self-catalysis is a prerequisite for life, considerably less attention has been devoted to network-based mutual-catalysis and its effect on evolution. To remedy this, we have used the graded autocatalytic replication domain (GARD) model, previously shown to capture essential features of reproduction, mutation and evolution in compositional molecular assemblies. We simulated a large ensemble of GARD rate-enhancement networks, thus allowing one to better study the crucial network properties of the implicated molecular assemblies. We found, with high statistical power, that high prevalence of mutual-catalysis is required for the emergence of appreciable diversity and evolvability of the assemblies, as well as for them to have significant selection attributes. We suggest that only minimal self-catalysis capabilities are needed to facilitate evolution-like behavior, and that excess self-catalysis may drive a population towards an evolutionary 'dead-end'.

## Introduction

A metabolism-first scenario for the origin of life entails that as early as replicating entities have emerged in the prebiotic soup, they must have constituted relatively complex molecular networks, arising via spontaneous accretion of early assemblies of simpler organic molecules (Dyson 1982; Bachmann et al. 1992; Kauffman 1993; Segre et al. 1998a; Luisi et al. 1999; Szathmáry 2000; Segre et al. 2001a; Shapiro 2006; Barandiaran and Ruiz-Mirazo 2008). In this scenario it is further proposed that faithful assembly reproduction directly stems from certain network attributes. To provide support for this scenario one must better understand the network properties of the implicated molecular assemblies. The GARD kinetic model for origin of life describes the homeostatic growth and evolution of an assembly composed out of a repertoire of  $N_G$  simple organic molecules (Segre et al. 1998a; Segre et al. 1998b; Segre et al. 2000; Segre et al. 2001a; Segre et al. 2001b; Shenhav et al. 2003; Shenhav et al. 2005; Hunding et al. 2006; Lancet et al. 2006) typically assumed to consist of amphiphilic molecules, e.g. lipids, and suggests a possible pathway to the formation of a minimal protocell (Shenhav et al. 2003; Szathmáry et al. 2005; Lancet et al. 2006; Thomas and Rana 2007; Chen and Walde 2010). The model is based on a catalytic network,  $\beta$ , usually presented in the form of a non-symmetric  $N_G \times N_G$  matrix, and

the system is kept away from equilibrium by imposing a fission action once an assembly reaches a size of  $N_{\max}$ . Key in GARD are compotypes – clusters of replication-prone quasi-stationary states (composomes) appearing during GARD dynamics – that make the compositional-genome (Segre et al. 2000; Segre et al. 2001b; Lancet et al. 2002; Shenhav et al. 2007) – and take an essential role in evolutionary processes related to GARD. Here, we used GARD simulations to ask how attributes of the mutually catalytic network embodied in the  $\beta$  matrix govern the evolution-related dynamics of compositional assemblies. We further report that GARD compotypes may display appreciable selection, contrary to a recent report (Vasas et al. 2010).

## Simulations

The model is subjected to a kinetic Monte-Carlo simulation based on Gillespie's algorithm (Gillespie 1976; Gillespie 1977; Segre et al. 1998a; Segre et al. 1998b), using parameter values similar to those typically employed in previous studies. A set of 10,000 GARD instances is generated, each with the same parameters and a different  $\beta$  matrix generated by a randomization of the same lognormal distribution (with a mean value of -4 and standard deviation of 4). Such random sampling of the  $\beta$  mutual catalysis network may be perceived as representing different possible GARD environmental chemistries.

## Compotypal diversity

Fig. 1 shows the correlation between the number of compotypes and the self-catalysis power (Eq. 1). As the propensity of self-catalysis increases, the probability of networks exhibiting a higher number of compotypes ( $>3$ ) is dramatically reduced,

$$p_{SC} = \frac{\sum_{q=1}^{N_G} \beta_{qq}}{\sum_{i=1}^{N_G} \sum_{j=1}^{N_G} \beta_{ij}} \quad (1)$$

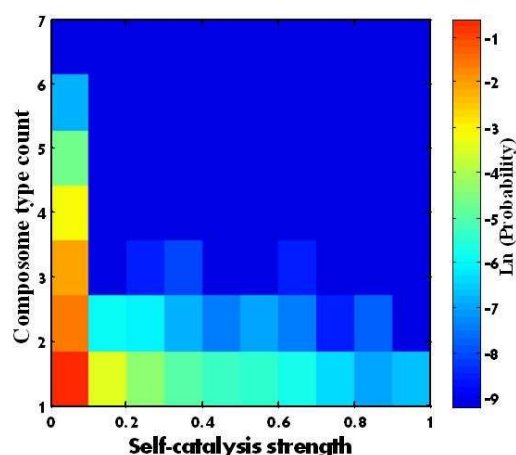


Figure 1: A density plot for the correlation between the number of composites and the self-catalysis power, obtained from 10,000 GARD instances.

meaning fewer possible targets for selection and therefore possibly hindering the selection response. Curiously, even among the majority of simulations that show only one composite, a large portion has low self-catalysis strength, suggesting that low self-catalysis strength is a necessary but not sufficient condition for high number of composites.

## Selection in GARD

In order to assess the selection response of GARD assemblies, a 'selection-GARD' simulation is performed by choosing a target composite and then running the simulation while temporarily biasing the growth of assemblies towards that target, based on a slight fitness gain and level of compositional similarity between an assembly and the target. A 'selection excess' (SE) parameter is subsequently defined, by comparing the frequencies of the target composite before and after selection. An increase in the target frequency as a response to selection pressure means positive selection and is represented by  $SE > 1$ . Similarly  $SE < 1$  and  $SE = 1$  represents negative and no selection, respectively.

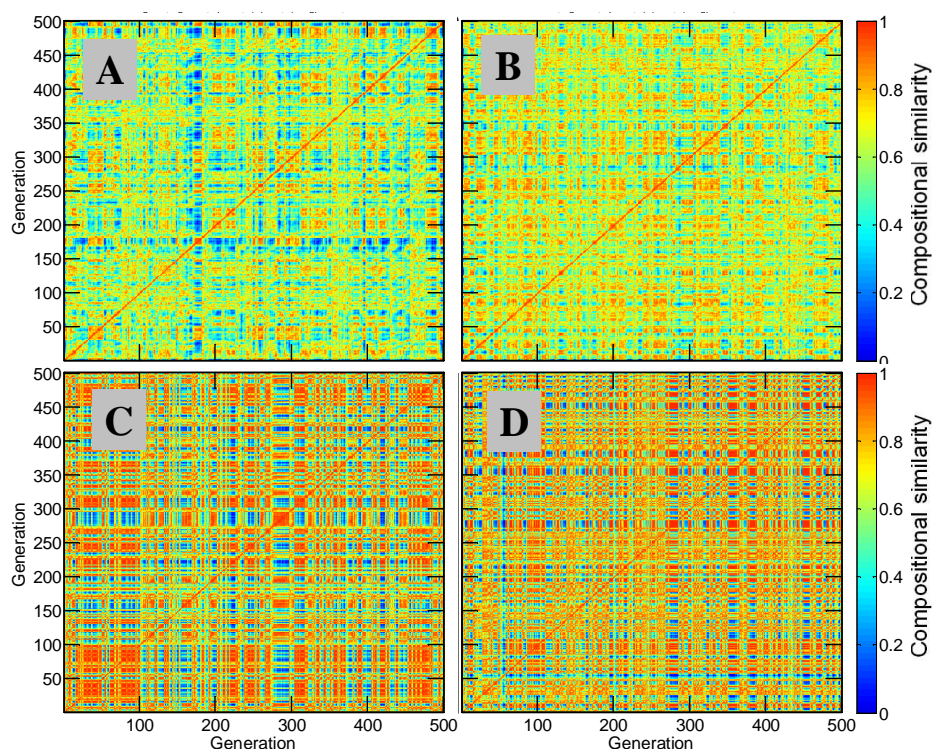


Figure 2: Examples of 'similarity carpet' before (A, C) and after (B, D) selection. The first example (A, B) is of a simulation positively responding to selection, as seen by the frequency of the target composite increasing by 75% ( $SE = 1.75$ ) in response to the selection pressure, seen as higher preponderance of large overall compositional similarity. The second example (C, D) is of negative selection. The frequency of the target is diminished ( $SE = 0.83$ ) seen as lower preponderance of large overall compositional similarity

We used the parameter SE to assess the capacity of GARD assemblies to undergo a process akin to Darwinian selection. Figs. 2 show examples of how GARD assemblies positively and negatively respond to selection pressure. Analysis of the entire 10,000 simulation instances reveals that a considerable percentage (~30%) of the networks show positive response to selection ( $SE > 1$ , and as large as  $SE = 2$ ). The mean SE value in this range is found to be about 1.4.

Such observations are contrary to a recent report (Vasas et al. 2010), where a claim has been made that specific GARD compositions show a negligible capacity to respond to selective pressure. Moreover, we interestingly find that networks with strong self-catalytic power exhibit practically no selection, and that the same range of low self-catalysis power that allows for high compotypal diversity also displays selection (both positive and negative), suggesting that GARD  $\beta$  networks must have an optimal ratio of self- to mutual-catalysis to manifest effective evolution-like behavior.

## References

- Bachmann, P. A., Luisi, P. L. and Lang, J. (1992). Autocatalytic Self-Replicating Micelles as Models for Prebiotic Structures. *Nature* 357(6373): 57-59.
- Barandiaran, X. and Ruiz-Mirazo, K. (2008). Modelling autonomy: Simulating the essence of life and cognition. *Biosystems* 91(2): 295-304.
- Chen, I. A. and Walde, P. (2010). From Self-Assembled Vesicles to Protocells. *Cold Spring Harbor Perspectives in Biology* 2(7): -.
- Dyson, F. J. (1982). A Model for the Origin of Life. *Journal of Molecular Evolution* 18(5): 344-350.
- Gillespie, D. T. (1976). General Method for Numerically Simulating Stochastic Time Evolution of Coupled Chemical-Reactions. *Journal of Computational Physics* 22(4): 403-434.
- Gillespie, D. T. (1977). Master Equations for Random-Walks with Arbitrary Pausing Time Distributions. *Physics Letters A* 64(1): 22-24.
- Hunding, A., Kepes, F., Lancet, D., Minsky, A., Norris, V., Raine, D., Sriram, K. and Root-Bernstein, R. (2006). Compositional complementarity and prebiotic ecology in the origin of life. *Bioessays* 28(4): 399-412.
- Kauffman, S. A. (1993). The origins of order: Self organization and selection in evolution, Oxford University Press, USA.
- Lancet, D., Kafri, R. and Shenhav, B. (2002). Compositional genomes: pre-RNA information transfer in mutually catalytic assemblies. *Geochimica Et Cosmochimica Acta* 66(15A): A429-A429.
- Lancet, D., Solomon, A., Kafri, R. and Shenhav, B. (2006). The simplest cellular life. *Origins of Life and Evolution of the Biosphere* 36(3): 213-214.
- Luisi, P. L., Walde, P. and Oberholzer, T. (1999). Lipid vesicles as possible intermediates in the origin of life. *Current Opinion in Colloid & Interface Science* 4(1): 33-39.
- Segre, D., Ben-Eli, D., Deamer, D. W. and Lancet, D. (2001a). The lipid world. *Origins of Life and Evolution of the Biosphere* 31(1-2): 119-145.
- Segre, D., Ben-Eli, D. and Lancet, D. (2000). Compositional genomes: Prebiotic information transfer in mutually catalytic noncovalent assemblies. *Proceedings of the National Academy of Sciences of the United States of America* 97(8): 4112-4117.
- Segre, D., Lancet, D., Kedem, O. and Pilpel, Y. (1998a). Graded autocatalysis replication domain (GARD): Kinetic analysis of self-replication in mutually catalytic sets. *Origins of Life and Evolution of the Biosphere* 28(4-6): 501-514.
- Segre, D., Pilpel, Y. and Lancet, D. (1998b). Mutual catalysis in sets of prebiotic organic molecules: Evolution through computer simulated chemical kinetics. *Physica A* 249(1-4): 558-564.
- Segre, D., Shenhav, B., Kafri, R. and Lancet, D. (2001b). The molecular roots of compositional inheritance. *Journal of Theoretical Biology* 213(3): 481-491.
- Shapiro, R. (2006). Small molecule interactions were central to the origin of life. *Quarterly Review of Biology* 81(2): 105-125.
- Shenhav, B., Oz, A. and Lancet, D. (2007). Coevolution of compositional protocells and their environment. *Philosophical Transactions of the Royal Society B-Biological Sciences* 362(1486): 1813-1819.
- Shenhav, B., Segre, D. and Lancet, D. (2003). Mesobiotic emergence: Molecular and ensemble complexity in early evolution. *Advances in Complex Systems* 6(1): 15-35.
- Shenhav, B., Solomon, A., Lancet, D. and Kafri, R. (2005). Early systems biology and prebiotic networks. *Transactions on Computational Systems Biology I*: 14-27.
- Szathmáry, E. (2000). The evolution of replicators. *Philosophical Transactions of the Royal Society of London Series B-Biological Sciences* 355(1403): 1669-1676.
- Szathmáry, E., Santos, M. and Fernando, C. (2005). Evolutionary potential and requirements for minimal protocells. *Topics in Current Chemistry* 259: 167-211.
- Thomas, J. A. and Rana, F. R. (2007). The influence of environmental conditions, lipid composition, and phase behavior on the origin of cell membranes. *Origins of Life and Evolution of Biospheres* 37(3): 267-285.
- Vasas, V., Szathmáry, E. and Santos, M. (2010). Lack of evolvability in self-sustaining autocatalytic networks constraints metabolism-first scenarios for the origin of life. *Proceedings of the National Academy of Sciences of the United States of America* 107(4): 1470-1475.

# Polyethism in a colony of artificial ants

Chris Marriott<sup>1</sup> and Carlos Gershenson<sup>1</sup>

<sup>1</sup> IIMAS, Universidad Nacional Autónoma de México, México City, México  
algorithm0r@gmail.com  
cgg@unam.mx

## Abstract

We explore self-organizing strategies for role assignment and strategy selection in a foraging task carried out by a colony of artificial agents. Foraging strategies are selected by strategies inspired by various mechanisms of division of labor (polyethism) observed in eusocial insects like ants, termites, or bees. Specifically we instantiate models of *caste polyethism* and *age* or *temporal polyethism* to evaluate the benefits to foraging in a dynamic or unknown environment. We focus on the ability of division of labor mechanisms to self-organize individual strategy selection based on the environment.

## Introduction

The self-organizing strategies of eusocial insects are now well known and well studied in biology (Beckers et al. (1989); Traniello (1989); Robinson (1992); Theraulaz et al. (1998); Theraulaz and Bonabeau (1999); Gautrais et al. (2002); Roulston and Silverman (2002); Merkle and Middendorf (2004); Garnier et al. (2007)) and applications to computation are abundant (Bonabeau et al. (1999); Panait and Luke (2004b,a); Schmickl and Crailsheim (2008); Gershenson (2010); Ducatelle et al. (2010)). One of the more remarkable behaviors observed is the ability of rather simple, unintelligent agents (individual insects) to coordinate their behavior to establish a rather fluid and adaptive behavior on the colony level. The phenomenon of *stigmergy* (communication via the environment) has now been modeled and applied in artificial simulations to achieve similar results among rather simple artificial agents (Theraulaz and Bonabeau (1999); Bonabeau et al. (1999); Panait and Luke (2004b,a); Schmickl and Crailsheim (2008)) cooperating in multi-agent systems.

However, many of these applications focus on homogeneous colonies, where each agent has the same behavioral capabilities. Nonetheless, observations of insects show that in many colonies the individuals are not always homogeneous. Colonies consist of heterogeneous agents, whether these agents display morphological differences (i.e. distinct castes) or merely behavioral differences. The effects of this

stratification of agents in a colony is referred to as division of labor (DOL) or by the term *polyethism* (Robinson (1992); Traniello and Rosengaus (1997); Theraulaz et al. (1998); Gautrais et al. (2002); Gordon (2003); Merkle and Middendorf (2004)). As artificial multi-agents systems grow larger and involve agents with different roles the problem of assigning roles to agents becomes increasingly important (Campbell and Wu (2010); dos Santos and Bazzan (2009)).

Biologists differentiate between at least two means of dividing roles amongst workers in natural insect colonies. The means we select for study are called *caste polyethism* and *age polyethism*. Other types of polyethism are also observed (e.g. *elitism*) and the two above types have many possible underlying mechanisms though these additional types and subtypes will not be explored in detail in this article. Simulations have just begun exploring task assignment and heterogeneous agent populations (e.g. Schmickl and Crailsheim (2008); Ducatelle et al. (2010)). Our experiment differs from these in that our agents are assigned the same task (foraging), but must decide which strategy to adopt to solve the task (between an individual exploratory strategy and a cooperative exploitative strategy). Other experiments focus on simulations of actual natural colony behavior in an attempt to assess models of those behaviors, whereas while we are inspired by these models our focus is on polyethism as a self-organizing strategy selection mechanism.

## Polyethism

*Caste polyethism* occurs when distinct types of individuals are bred by the colony. An individual is effectively born into its role, often times displaying morphological differences from individuals from other castes. The clearest example of castes is the division between the reproductive caste and the worker caste in eusocial insects. A single or small group of reproductive females (called queens) are responsible for all reproductive tasks in the colony while non-reproductive workers carry out all other tasks required by the colony (brood care, nest constructions and maintenance, waste removal, foraging, and defense). In some species workers are further divided into sub-castes. Differences among workers

from different castes are particular to the worker's role. For instance in some species of ants the workers can be divided into *majors* and *minors* (occasionally with an intermediate caste as well) where the majors are larger than the minors, this size being helpful in the task they carry out (primarily colony defense). Minors are smaller, making them more energy efficient, and they are relegated to less dangerous tasks like foraging and nest maintenance. Only in rare occasions will a worker do a task that is typically assigned to a different caste.

*Age* or *temporal polyethism* is a type of division of labor where the worker's role is correlated with its age or changes over time. Age polyethism is more common than caste polyethism in natural insect colonies. In colonies displaying age polyethism younger workers commonly carry out less risky tasks (nursing or nest maintenance allowing them to stay in the nest) whereas older workers carry out more dangerous tasks (foraging, defense, or raiding where the agent must leave the nest). It is hypothesized that this division of labor allows the colony to maximize the work carried out by each individual worker (i.e. young workers will be less likely to die and thus can live longer to carry out more work). This will be beneficial to the colony since it will have to breed fewer workers if each worker's longevity (and thus productivity) is maximized (Tofilski (2002, 2009)). In certain cases this progressive role assignment may also allow younger and less experienced workers to gain the experience necessary to carry out more difficult tasks (say at the very least allowing them to become familiar with the layout of the nest and surrounding environment before having to venture far from the nest) (Tofts and Franks (1992); Tofts (1993); Franks and Tofts (1994)). Many mechanisms have been suggested as the underlying reason for observed age polyethism. The mechanism we employ is similar to the response threshold model commonly studied (see e.g. Theraulaz et al. (1998); Garnier et al. (2007)).

## Artificial Ants

The experiment detailed below involves a colony of artificial ants engaged in a foraging task. The colony level task is to maximize the food intake of the colony (allowing colony sustenance and growth). On the individual worker level the task is to explore the environment, find a food object, and return to the nest with the object.

We consider two different strategies for individual workers inspired by natural ant populations. The first, and simpler, strategy is for workers to forage for the most part *individually*. We say "for the most part" here since individual foragers cooperate at least insofar as they attempt to divide the environment to be explored equally among them (see Figure 1). We implement this strategy by having ants leave a "seeker" trail as they leave the nest. While "seeking" the ants will avoid other seeker trails, meaning they will travel away from the nest while avoiding the trail they leave be-

hind them, but they will also avoid trails left by other ants, helping to divide the area somewhat evenly. Other than this simple cooperation, workers leave the nest and randomly explore until they find a food object (or reach the range of their exploration) and return to the nest. We will call this strategy the "individual" or "exploratory" strategy, and ants following this strategy "explorers". The seeker path left by these ants also serves as the ants' sole means of returning to the den (i.e. they follow seeker paths back).

Ants that find a food source of sufficient size (i.e. they find at least one food morsel to carry back to the nest and at least one more food morsel they will recruit others to seek out) the ant will leave a second type of trail we call the "carrier" trail. The second strategy, which we call the "cooperative" strategy or "exploitative" strategy, involves foragers that will follow "carrier" trails to exploit food sources that were already discovered by other ants. Both explorers and exploiters will leave "carrier" trails under the conditions listed above, but only exploiters will follow them to food sources.

Trails in our simulation consist of discrete pheromones. The trail to be followed is selected randomly from observed trails with probability weighted relative to the trail's decay (newer trails more likely to be followed than older ones). Trails to be avoided are all considered in avoidance behavior, however if there is a trail to follow the avoidance behavior is suppressed.

These strategies are inspired by those found in natural populations, with a correlation of colony size to the strategy used (Beckers et al. (1989); Traniello (1989)). It has been observed that smaller colonies tend to use the exploratory strategy whereas as the larger the colony is the more likely the colony uses an exploitative strategy. Despite this correlation, upon closer examination larger colonies have foragers carrying out both strategies, that is, they engage in polyethism.

It is known (Roulston and Silverman (2002)) that these strategies fare differently depending on the environment the colony is situated in. If food objects are *uniformly* distributed around the nest then the individual strategy reaches near optimal foraging. Over time the workers will clear a disc shaped area of food around the nest, the radius of the disc being determined by the frequency of food objects and by the size of the population. This situation is presented in Figure 1.

Interestingly, in larger colony sizes the cooperative strategy also fares quite well in environments with uniform distribution of food, though the foragers carry out a more complex foraging strategy. Cooperative foragers form an "arm" leading from the nest into the environment and this arm has been observed to swing in a circle around the nest, clearing food objects as it goes, or spontaneously dissolving and reforming in a more lucrative direction. These strategies have also been observed in natural ant colonies. While the cooperative strategy seems to approach the performance of the in-



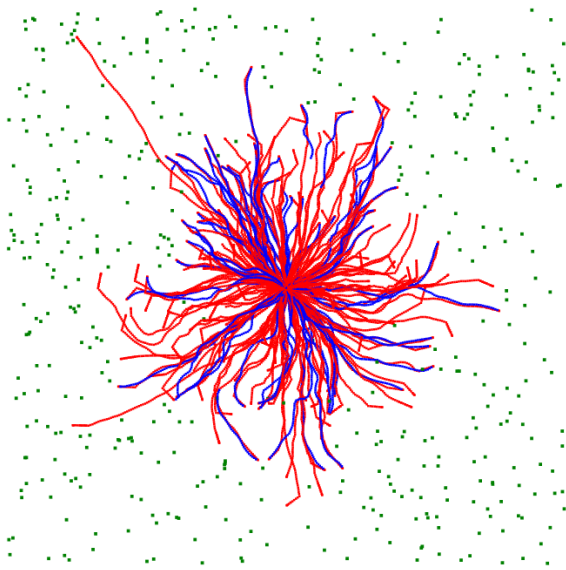


Figure 1: Explorers in a uniform environment. The den is in the center of the torus. Green squares are food. Red paths are seeker paths. Blue paths are carrier paths. Recall that explorers ignore the carrier paths.

dividual strategy in experimentation, the individual foragers have an advantage in an environment with uniformly distributed food.

A second environment type we have investigated contains food isolated in *patches*. For the sake of comparison among simulation runs our food patches are always placed equidistant from the nest, though in a random direction. In this environment the cooperative foragers have a clear advantage. Once a forager finds a patch of food it recruits other foragers to help it clear the patch and the colony quickly optimizes the path to the food patch. Figure 2 shows a typical patch environment (with 2 patches) and a colony of exploitative ants foraging from the patches.

Individual foragers are at a significant disadvantage when faced with an environment with a single patch. Many individual foragers leave the nest in the wrong direction and return empty handed.

Given the differential success of these strategies in these environments it is our hypothesis that polyethism in a colony will be beneficial if the colony is faced with either an unknown environment (of one of these two types) or with a dynamic environment consisting of either a combination of these types or shifting between these types.

## Experimental Setup

In our experiment we consider four different types of colonies that we will expose to five different types of environment. We will consider how each colony fares in each environment, as well as how the colony fares across all en-



Figure 2: Exploiters in an environment with two patches. The den is in the center of the torus. Green squares are food. Red paths are seeker paths. Blue paths are carrier paths. Exploiters use the carrier paths to cooperatively forage.

vironments.

A colony will consist of a *queen*, a population of *workers*, a population of *larvae*, and a store of food. Workers consume food at a constant rate (about 1 food every 450 simulation rounds) and larvae consume food at a constant rate (1 food for the 100 round gestation period).

The queen lives for the duration of the experiment (or until the colony dies of starvation), though workers and larvae may die. Workers die under two conditions. If they reach their maximum age (selected uniformly from the range 2750-3250 rounds), or if they run out of food energy. When a worker consumes a piece of food it gains energy that will sustain it for 450 simulation rounds. If while foraging the worker's food energy reaches 0 (i.e. after 450 rounds) then the worker attempts to return to the nest (possibly without food). Upon returning the worker will attempt to consume a unit of food from the store. If there is no food in the store the worker dies.

A larvae also consumes food, once upon creation by the queen and again upon changing into a worker. The food consumed when the larvae matures forms the initial energy store of the worker. A queen will never create a larvae in an instance where the food stores are empty, however, a larvae may mature and find the store empty. In this case the new worker dies.

Queens from different colonies have different profiles, however, they all follow the same rule when deciding to reproduce. A queen will only create a new larvae if the food store exceeds the current population of workers plus the cur-



rent population of larvae.

## Colony Types

The first two types of colony will form a control group for comparison. These two types will not use polyethism and queens in these colonies will create *only explorers* or *only exploiters* respectively. From the earlier discussion we know that these colonies will fare well in some environments but not in others and will not be adaptive to a dynamic or unknown environment.

The third colony will engage in an *adaptive caste polyethism*. Queens in this type of colony produce larvae that can mature into either an individual or cooperative worker. The queen chooses the type of worker to create in proportion to the success rate of workers of that type. (The queen keeps track of food returned by each type of forager over the last 500 rounds, and of the number of each type of forager. From this she estimates the efficiency of the average ant of each type and randomly selects to create a new ant in proportion to the ratio of success rate.) Thus if explorers are more successful at foraging than exploiters then a queen will make an explorer with higher probability (and vice versa). Queens in this type of colony will ensure there is always at least one worker of each type so success rates can be properly estimated.

The fourth colony will engage in one type of *temporal polyethism*. Workers in these colonies are homogeneous in their behavioral repertoire, in that they can act as either explorers or exploiters. Which role a worker adopts depends first on their age (for younger workers) and then on the demands of the colony (for older workers). In this colony new workers adopt an individual foraging strategy, and may switch to a cooperative strategy (or back again) after reaching a particular age (usually consisting of 1 or 2 full foraging trips). Workers of this type choose to change roles based on collective experience, that is, in proportion to the success rate of workers in the colony similar to the mechanism used in the third colony.

(The estimation of the success rate of each strategy is carried out with simple counters in our simulation, though we believe these correspond to a basic stigmergic strategy. While we do not use pheromones in our model of this behavior we believe this mechanism is closely related to response threshold models of behavior selection.)

## Environmental Types

We expose these 4 colony types to 5 distinct environments: *uniform*, *patch*, *roaming patch*, *seasonal*, and *mixed*. The rate at which food drops in each environment is the same (1 food every 5 rounds) and each food will stay in the environment for exactly 1000 rounds or until picked up by a forager. The *uniform* and *patch* environments were described above consisting of uniformly distributed food or an isolated patch of food respectively.

The *roaming patch* environment has a single patch but this patch will change location every 1000 rounds (the new location will be the same distance from the nest as the old location). This means that after the patch has moved new food will drop in the new patch location, though old food is not removed unless foraged or it reaches its 1000 round limit. As a result there will usually be two patches in the environment, one containing old food that is decaying and one containing new food. Figure 2 displays a typical scenario for this type of environment.

The *seasonal* environment is intended to simulate an environment that changes from a uniform distribution to an isolated patch with regularity possibly corresponding to the seasons. We simulate this idea by alternating between the two distributions every 1000 rounds. Again there will be a temporal overlap between these two environments meaning that the environment will typically contain food dropped uniformly and in a patch. Every time the season changes to the patch distribution a new location for the patch is selected so in this sense we see the patch as roaming as in the last environment.

The *mixed* environment includes both uniform food drops and an isolated patch at the same time, and the environment is static (in that the patch does not move). In this environment the drop rate is the same as previous environments despite there being two active food drop mechanisms operating simultaneously.

## Observations and Data

We choose to analyze the worker population data of our colonies. This data reflects the colonies' ability to forage for food efficiently. Each colony begins with an initial food store of 32 food and zero ants. The queen will use this initial food to create 16 initial workers which mature on rounds 101-116 of the simulation. At this point the initial food stores will be exhausted. Parameters of the simulation determine a maximum colony size, namely the food drop rate and the energy consumption rate of the workers (as well to a lesser extent the size of the environment). This maximum is just above 80 workers, though due to the non-linear dynamics of the simulation this maximum can be exceeded for short periods. We used these parameters to balance computational time with robust results. A single run with a higher population suggests that the results scale, but further work would need to be done to verify this.

The initial stages of the simulation are occupied by rapid growth of population as the foragers are able to bring in more food than the colony needs so new workers are created (exceptions to this are noted below). This rapid growth commonly results in too many workers and so is often followed by a large dip in population and an oscillation is observed until an equilibrium can be found. This equilibrium depends on the type of colony and environment.

Figure 3 (left column) displays the worker population data

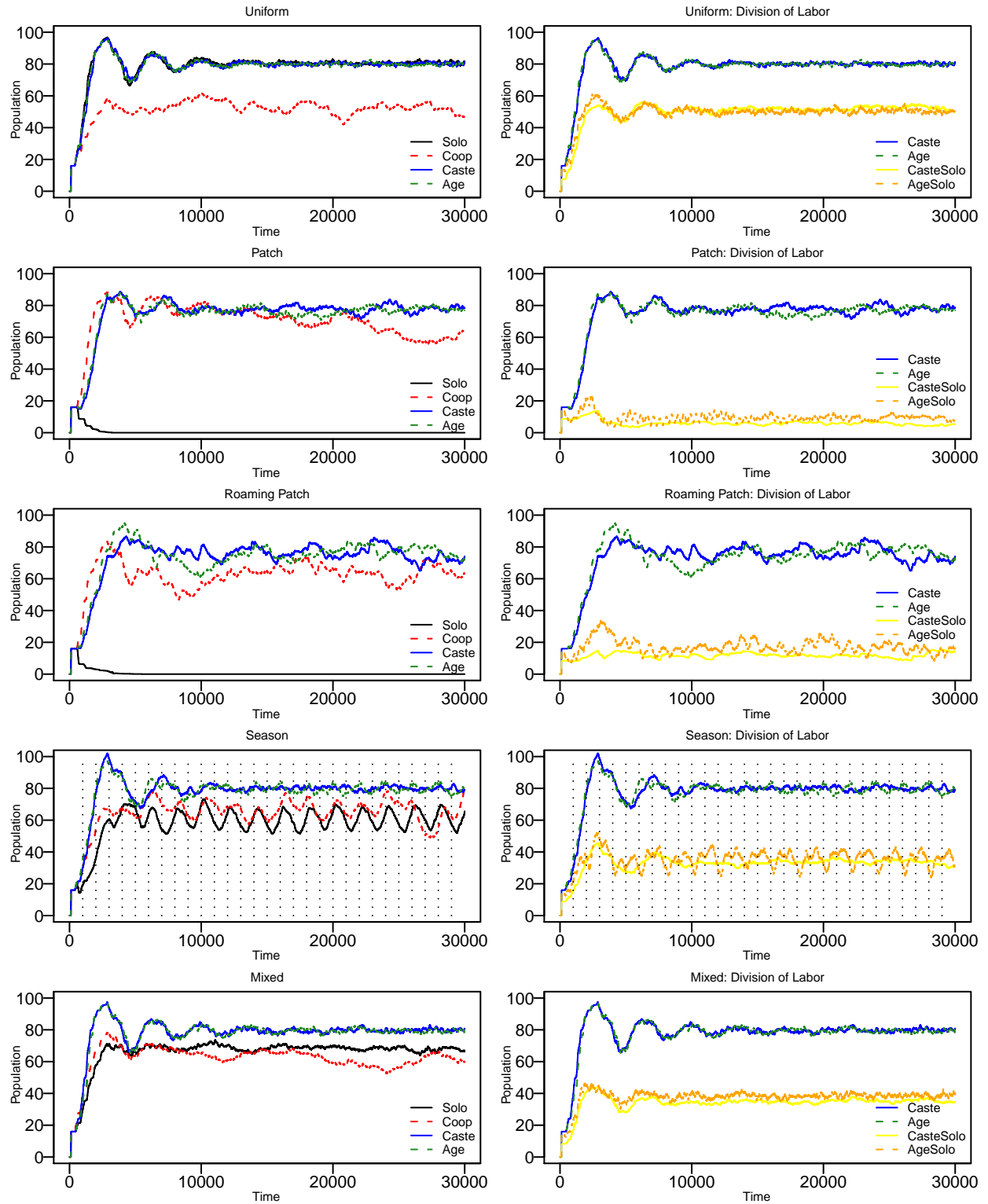


Figure 3: Worker Population Data. From the top row the data is presented for each environment: *uniform*, *patch*, *roaming patch*, *seasonal*, and *mixed*. The left column displays worker population over time for the four colony types. The right column displays the division of labor in the *Caste* and *Age* colonies. The worker population of these colonies is contrasted to the number of workers in the colony assigned to the *exploration* task. Please note we use "Solo" to indicate explorers and "Coop" to indicate exploiters in the charts.

gathered from all experimental runs. The data presented in the figure is the *average worker population* over time for  $N$  different simulation runs ( $N = 13$ ).

## Population Analysis

In the *uniform* environment the best performance is achieved by the explorers, and is closely matched by the caste and age polyethistic colonies. All three colonies settle around a population of 80 workers after initial instability. While the exploitative colony has no trouble surviving in this environment its sub-optimal foraging strategy allows it to maintain only a population of between 40-60 workers. Its population is also subject to greater instability as the foraging arm grows and shrinks in size and changes location. This is further supported by a greater average standard deviation (11.06) for exploitative population data compared to explorers, caste and age polyethistic colonies (2.37, 2.76, 2.71 respectively).

In the *patch* environment we again see expected results. The explorers are unable to maintain even the low initial colony size and the colony starves quickly. The cooperative foragers are the quickest to exploit the patch, whereas the polyethistic colonies are able to quickly adapt to the environment by producing exploiters instead of explorers. Both polyethistic colonies still maintain a small population of explorers. The dip in cooperative population observed near the end of the simulation is caused by two anomalous colonies from the simulation runs that starved to death. No such extinctions were observed among the polyethistic colonies. We observe some population instability in this environment. The average standard deviation for explorers (18.34) was about double that of the caste and age polyethistic colonies (9.13, 9.27 respectively) indicating greater stability from polyethism in this environment.

In the *roaming patch* environment we see that the polyethistic colonies are able to maintain a higher population than the purely exploitative colony (the explorers quickly starve in this environment as well). This implies a better ability to adapt to the moving patch. The exploitative colony also displays a greater instability in population though all three successful colonies have greater instability than in the stationary patch environment (supported by higher average standard deviations 22.13, 13.73, 13.25 for the exploiters, caste and age polyethistic colonies respectively) however still with greater stability in the polyethistic colonies. Also noteworthy is that all colonies have trouble maintaining an optimal population.

In the *seasonal* environment we again observe better performance from the polyethistic colonies than the purely explorer and purely exploiter colonies. Further there is greater stability of population in the polyethistic colonies (average standard deviations 7.76, 7.24 for caste and age respectively), where the pure explorer and pure exploiter colonies suffer population oscillations corresponding roughly to the

changing seasons (average standard deviations 13.77, 18.65 respectively). Note in the figures the dotted lines display the changing seasons. The polyethistic colonies manage to maintain roughly optimal populations in this environment while the explorer colony suffers the most in the seasons when food becomes isolated in a patch.

Finally, in the *mixed* environment, we again see a population advantage to polyethism. While both the purely explorer and purely exploiter colonies survive in the mixed environment they are unable to reach the optimal populations and display a slightly greater instability. The purely explorer population also maintains a slight population advantage over the purely exploiter population. Stability is highest in the explorer and polyethistic colonies (average standard deviations for this environment are 4.26, 15.74, 5.04, and 3.16 respectively).

## Division of Labor

A secondary focus of our simulations was on the division of labor in the polyethistic colonies. We gathered data on how many workers of each type were deployed at a given time by the polyethistic colonies. This data is presented in Figure 3 (right column) for each environment. We display only the number of explorer workers in the chart in contrast to the total worker population, with the number of exploitative workers being the difference. In the caste polyethism colonies this corresponded to how many workers of each caste were available. In the age polyethism colonies this corresponded to how many workers were currently assigned to each strategy during that round.

In the control environments the polyethistic colonies stabilized around a constant number of explorers. For the *uniform* environment both colonies settled at just over half of the workers (about 50 out of 80 workers) dedicated to exploring. It is worth noting that the colonies did not try to maximize the number of explorers in this environment. In the *patch* environment the caste colony settled at around 5 workers dedicated to exploring while the age colony maintained a slightly higher number of explorers, typically oscillating between 5 and 15 workers. We note that in these environments the age polyethistic colony displayed greater oscillations of worker assignments whereas the caste polyethistic colony tended to stabilize around a particular division of workers assigned to each task.

In the *roaming patch* environment more explorers were maintained than in the stationary patch environment. In the caste colony just over 10 of the workers were assigned the exploring role. The age colony still assigned more workers to exploring on average than the caste colony, typically above 15, but as high as 25. Again the age colony had greater variation in its division of labor.

The *seasonal* environment displays distinct performance differences between the two polyethistic colonies. The caste colony settles on 30 to 35 workers dedicated to exploring.

This number is stable when compared to the age colony which attempted to adjust the worker base to the current season. Thus we see the number of explorers oscillating between about 25 workers to as high as 43 workers (excepting the early spike).

In the *mixed* environment both polyethistic colonies stabilize their worker base by assigning roughly half the workers to each task. The age colony again assigns slightly more workers to exploration than the caste colony and displays slight oscillations.

## Discussion

The data presented suggests that polyethism, regardless of kind, offers benefits to the foraging task. While both of the foraging methods studied in this experiment (exploring and exploiting) can be seen as self-organizing methods, the colonies benefit if the “higher-level” self-organizing method of polyethism is applied to select which of the strategies to engage in (Gershenson (2010)). The clearest advantage shown by our experiment is the ability for these mechanisms of polyethism to *adjust the ratio* of explorers to exploiters based on the environment.

In the environments where the environment is specifically created to favor one of the two basic strategies, exploring or exploiting, we see that polyethism allows the colony to adjust the worker base to the environment. The only drawback in these environments to the polyethistic colonies is that they require some time to adjust to the environment.

In the more dynamic environments we see that polyethism is necessary to get optimal or near-optimal performance. We see that in the *roaming patch* environment, while exploiters are well suited for this environment, maintaining a small population of explorers allows the new patch location to be found quicker, and more quickly exploited (Roulston and Silverman (2002)).

In the *seasonal* and *mixed* environment polyethism is necessary to have optimal foraging. In the *seasonal* environment the non-polyethistic colonies suffer in seasons where they are not well suited. In the *mixed* environment the non-polyethistic colonies are unable to exploit all the food drops and thus cannot maintain as high a population. In the *mixed* environment the polyethistic colonies settle on a division of workers among the two strategies that allows for exploiting both food sources. It is interesting to note that the polyethistic colonies still managed to reach optimal population levels in the mixed environment, implying that the two strategies did not experience negative interference.

We also note that the polyethistic colonies adopt different approaches to the *seasonal* environment. The caste colony maintains a constant number of each worker. This can be seen as the colony being prepared for either season, but not necessarily specializing for the current season. This approach may be favored by the caste colony because the season length (1000) is short compared to the lifespan

of a worker (selected uniformly from the range 2750-3250 rounds). Thus the caste colony will not have the opportunity to adjust the balance of workers each season since workers from the previous season will still be present in the work force. The age colony *does* adjust its work force to the new season, albeit only slightly, since the workers in this colony can switch tasks every round trip which is about 300-400 rounds long, shorter than the season length. Both strategies allow the colony to maintain fairly stable and nearly optimal populations.

To test this analysis we conducted a follow up experiment where the season size was extended to 3000 rounds (see Figure 4). In this run we saw that the caste colony adopted the adjustment strategy as well, attempting to match workers to the season instead of opting for an equal distribution. We observed in this case that the age colony was able to adapt its workers more rapidly than the caste colony, and thus had a slightly more stable population. The stability of both colonies' populations suffered with the longer seasons due to more polarization of the workforce and the lag between the season change and the ability of the colony to adjust its workforce.

## Conclusion

We conclude that *division of labor* is beneficial to ant colonies in that it adds a layer of dynamism to their problem solving as well as makes the colony more robust. We suggest that the simple self-organizing methods of assigning workers to tasks can be adopted in artificial systems. These methods are simple to implement and require a minimal amount of central planning or control. The methods are reactive and dynamic and can likely be applied in a variety of situations, this being the topic of future work.

While we found little evidence favoring one of age or caste polyethism as a method of assigning workers to tasks we did find that the caste polyethism appeared to be more rigid in that it took longer for the workforce to adjust to new conditions. However, the trade off is that in the age polyethistic colonies there was a tendency to over adjust to new conditions, which may not be favorable in all situations. We believe that more work is required to determine the benefits of each of these methods, given that the distribution of these methods among natural colonies is not balanced (recall age polyethism is more common than caste polyethism). One aspect that was not considered in this experiment, and probably plays an important role in natural colonies, is the variable costs to a colony or species (genetically and in terms of energy expenditure) in producing workers that either are specialized for their task (caste polyethism) or are generalists able to take on any available task (age polyethism).

## References

- Beckers, R., Goss, S., Deneubourg, J., and Pasteels, J. (1989). Colony size, communication, and ant foraging strategy. *Psy-*

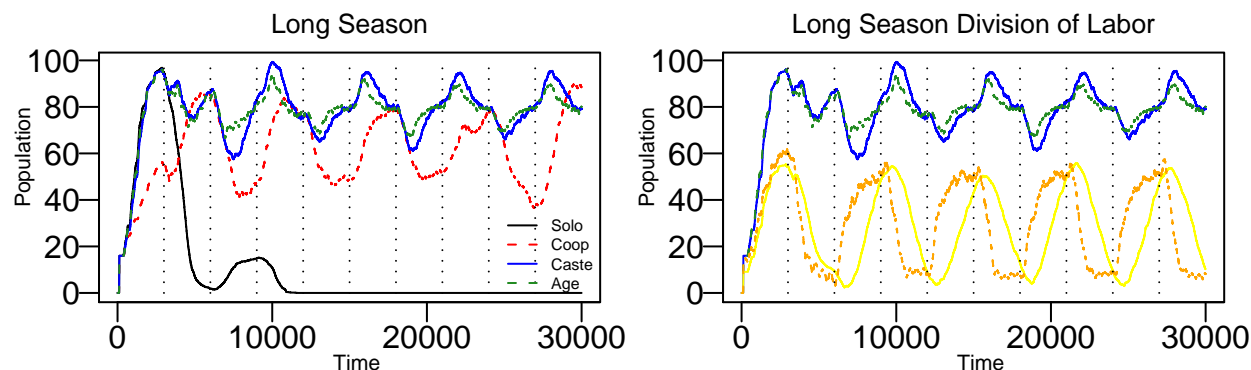


Figure 4: Long Season (3000 round). The left chart displays population over time for the four colony types. The right chart displays the division of labor in the *Caste* and *Age* colonies by contrasting the total population to the number of workers in the exploration task (the legend has been removed for clarity though we follow the same format as Figure 1).

che, 96:239–56.

Bonabeau, E., Dorigo, M., and Théralaz, G. (1999). *Swarm Intelligence: From natural to artificial systems*. Oxford University Press.

Campbell, A. and Wu, A. (2010). Multi-agent role allocation: Issues, approaches, and multiple perspectives. *Autonomous Agents and Multi-Agent Systems*, 22(2):317–355.

dos Santos, F. and Bazzan, A. (2009). An ant based algorithm for task allocation in large-scale and dynamic multiagent scenarios. In *Proceedings of the 11th annual conference on Genetic and evolutionary computation (GECCO'09)*.

Ducatelle, F., Di Caro, G., and Gambardella, L. (2010). Cooperative self-organization in a heterogeneous swarm robotic system. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation (GECCO'10)*.

Franks, N. and Tofts, C. (1994). Foraging for work: how tasks allocate workers. *Animal Behavior*, 48:470–472.

Garnier, S., Gautrais, J., and Theraulaz, G. (2007). The biological principles of swarm intelligence. *Swarm Intelligence*, 1:3–31.

Gautrais, J., Theraulaz, G., Deneubourg, J., and Anderson, C. (2002). Emergent polyethism as a consequence of increased colony size in insect societies. *Journal of Theoretical Biology*, 215:363–373.

Gershenson, C. (2010). Computing networks: A general framework to contrast neural and swarm cognitions. *Paladyn, Journal of Behavioral Robotics*, 1(2):147–153.

Gordon, D. (2003). The organization of work in social insect colonies. *Complexity*, 8(1):43–46.

Merkle, D. and Middendorf, M. (2004). Dynamic polyethism and competition for tasks in threshold reinforcement models of social insect. *Adaptive Behavior*, 12(3-4):251–262.

Panait, L. and Luke, S. (2004a). Ant foraging revisited. In *Proceedings of the ninth international conference on the simulation and synthesis of living systems (ALIFE 9)*.

Panait, L. and Luke, S. (2004b). Learning ant foraging behaviors. In *Proceedings of the ninth international conference on the simulation and synthesis of living systems (ALIFE 9)*, pages 569–574.

Robinson, G. (1992). Regulation of division of labor in insect species. *Annual Review of Entomology*, 37:637–65.

Roulston, T. and Silverman, J. (2002). The effect of food size and dispersion pattern on retrieval rate by the argentine ant, *Linepithema humile* (hymenoptera: Formicidae). *Journal of Insect Behavior*, 15(5):633–648.

Schmickl, T. and Crailsheim, K. (2008). Analysing honeybees' division of labour in broodcare by a multi-agent model. In *Proceedings of the 11th conference on the simulation and synthesis of living systems (ALIFE11)*.

Theraulaz, G. and Bonabeau, E. (1999). A brief history of stigmergy. *Artificial Life*, 5:97–116.

Theraulaz, G., Bonabeau, E., and Deneubourg, J.-L. (1998). Response threshold reinforcements and division of labour in insect societies. In *Proceedings of the Royal Society*, pages 327–332.

Tofilski, A. (2002). Influence of age polyethism on longevity of workers in social insects. *Behavior Ecology and Sociobiology*, 51:234–237.

Tofilski, A. (2009). Shorter-lived workers start foraging earlier. *Insectes Sociaux*, 56(4):359–366.

Tofts, C. (1993). Algorithms for task allocation in ants (a study of temporal polyethism theory). *Bulletin of Mathematical Biology*, 55:891–918.

Tofts, C. and Franks, N. (1992). Doing the right thing: ants, honeybees and naked mole rats. *Trends in Ecology and Evolution*, 7:346–349.

Traniello, J. (1989). Foraging strategies of ants. *Annual Review of Entomology*, 34:191–210.

Traniello, J. and Rosengaus, R. (1997). Ecology, evolution and division of labour in social insects. *Animal Behavior*, 53:209–213.

# Tipping the Scales: Guidance and Intrinsically Motivated Behavior

Georg Martius<sup>1</sup> and J. Michael Herrmann<sup>2</sup>

<sup>1</sup> Max Planck Institute for Mathematics in the Sciences, Inselstr. 22, 04103 Leipzig, Germany

<sup>2</sup> University of Edinburgh, School of Informatics, IPAB, 10 Crichton St, Edinburgh EH8 9AB, U.K.

`martius@mis.mpg.de, michael.herrmann@ed.ac.uk`

## Abstract

We propose a novel approach to learning in autonomous robots that relies on the dynamical maintenance of an actively sensitized sensorimotor loop. Very weak learning cues are sufficient to orient a robot towards the desired behavior which is then selected from the intrinsic exploratory movements rather than imposed by a control command. The learning paradigm is a form of guided self-organization and is complementary to both active and intrinsically motivated learning. We present a systematic analysis of the learning algorithm in a robot control task and demonstrate its remarkable scalability with respect to the degrees of freedom of the system.

## Introduction

Learning in autonomous agents implies an active involvement of the agent in the acquisition of new behavior. Lopez and Oudeyer (2010) ask for a unified formalism for active and intrinsically motivated exploration and observe a convergence of approaches from machine learning and developmental psychology towards a new perspective for developmental robotics. While a number of examples exist that impressively demonstrate the virtues of this view, it appears that a different sets assumptions are required that may eventually turn out to limit the possibility of on-going learning, scaling and transfer across domains. Since a more extended discussion is beyond the present scope we should mention here merely that the present approach aims at a relaxation of some of these assumptions. We will use only a local world model

While some variants of intrinsically motivated learning try to extract controllable options (Singh et al., 2004; Martius et al., 2008; Jung et al., 2011) we will use here a related approach (Martius and Herrmann, 2010) in order to improve the sensitivity with respect to given learning signals (cues). We implement in this way a form of self-organized curiosity (Schmidhuber, 1991; Herrmann, 2001) for the cues which substantially improves goal-related learning in an autonomous robot. We will show examples where the learning time within this approach scales very nicely with the complexity of the problem.

We start from an approach to self-organization of robot control (Der, 2001; Martius et al., 2011) which aims at robotic behaviors that are characterized by on-going exploration and that can be called natural for a specific robot in a particular environment (Der et al., 2006; Hesse et al., 2009). Animals, including humans, can be assumed to acquire their behavioral repertoire in a similar way: Behavioral elements are developed autonomously and are composed and refined later in order to realize more complex goals. The resulting behavior is, nevertheless, subject to an on-going developmental modulation throughout the whole life span.

In robotics, many promising examples for autonomous behavioral adaptation and generation have been studied for instance by Herrmann (2001); Tani (2003); Der et al. (2006); Nolfi (2006); Oudeyer et al. (2005). Self-organization of behavior is, nevertheless, still a field of active exploration. Further questions such as the interaction of learning by self-organization and learning by supervision or by external reinforcement are just starting to gain scientific interest.

Usually, goal-oriented behavior is achieved by direct optimization of the parameters of a control program such that the goal is approached more closely. The learning system must receive information about whether or not the behavior actually approaches the goal. This information may be available via a reward signal in reinforcement learning or by a fitness function in evolutionary algorithms. We will consider similar types of goal-related information when aiming at a combination of self-organizing control with external drives. For this combination the term *guided self-organization* (GSO) was proposed by Martius et al. (2007); Prokopenko (2009). In a general perspective, GSO is the combination of goal-oriented learning and developmental self-organization. Each of the two learning paradigms bring about their particular benefits and GSO aims at combining them in an optimal manner. For instance, self-organizing systems tend to have a high tolerance against failures and degrade gracefully, which is also desirable in task-oriented applications. when developing systems aiming to achieve tasks in practical applications. We will deal in with a specific approach to self-organizing control, namely homeokinetic learning.

*Homeokinetic learning* generates self-organized behavior which can serve as intrinsic motivation of the robot to become engaged with its environment. The robot learns to maintain an active low-level sensorimotor loop without abstract or specific information. Here we will study the possibility of including high-level information into this dynamical systems approach such that the robot can learn to reach a goal or to optimize its behavior according to external standards.

What can we expect from a *guided homeokinetic controller*? It has been shown earlier by Der et al. (2006) and Hesse et al. (2009) that a variety of behaviors can emerge from the principle of homeokinesis. The emerging behaviors show a coherent sensorimotor dynamics of the particular robot in its environment. With additional guidance the exploration of the homeokinetic controller can be channeled around desired or preferred behaviors such that control modes can be quickly found which match the given robotic task.

The behavior is essentially driven by intrinsic self-organization, while the goal is easily taken up by the system due to the optimal sensitivity of the homeokinetic control. In a sense, we are not considering here an approach to robot learning but rather an on-going dynamic realization of the (external or internal) hints as part of an exploratory regime.

In the present paper, we will advance our study of guided self-organization of behavior, presented in Martius and Hermann (2010), by an application to a high-dimensional system. In order to keep the paper self-consistent, we introduce the homeokinetic control principle in the next section and present then the guidance by supervised *teaching cues*. The latter are the basis for the guidance by cross-motor teaching that can be implemented by the specification of abstract motor relations. We will extend this framework and apply it to the locomotion of bracelet-like robots with up to 40 DoF.

## Self-Organized Closed-Loop Control

Self-organizing control for autonomous robots can be achieved by an intrinsic drive towards active and predictable behavior as described by the homeokinetic principle (Der, 2001). We assume that the dynamics of the sensor values  $x \in \mathbb{R}^n$  of the robot can be written as

$$x_{t+1} = \psi(x_t) + \xi_{t+1} \quad (1)$$

where  $\psi$  is the internal model maintained and adapted by the robot to predict future sensor values and  $\xi$  is the prediction error. The motor values (actions)  $y \in \mathbb{R}^m$  are generated by a controller implemented simply as a parametric map or one-layer neural network:

$$y_t = K(x_t, \mathcal{C}_t) = g(C_t x_t + h_t) \quad (2)$$

where  $g(\cdot)$  is a sigmoid function with  $g_i(z) = \tanh(z_i)$ . The controller parameters  $\mathcal{C}$  consist of a weight matrix  $C$  and a

bias vector  $h$ . We compose the map  $\psi$  from a forward model  $M(x, y, \mathcal{A})$  and the controller  $K(x, \mathcal{C})$  (Eq. 2) as

$$\psi(x_t) = M(x_t, y_t, \mathcal{A}_t) = M(x_t, K(x_t, \mathcal{C}_t), \mathcal{A}). \quad (3)$$

The function  $M$  is initially unknown, but the robot adapts it continuously in order to minimize the prediction error  $\xi_t$  by

$$\mathcal{A}_{t+1} = \mathcal{A}_t - \epsilon_a \frac{\partial}{\partial \mathcal{A}_t} \|\xi_t\|^2. \quad (4)$$

If the parameters  $\mathcal{C}$  were also adapted in this way then stable but typically trivial behaviors would be produced unless specific information is given to the robot.

The homeokinetic principle which we are going to use here normally does not need any specific information in order to produce a variety of elementary behaviors in a robot. We will show that this principle for the self-organization of behavior offers also a new perspective for learning in robots. That is, if additional information is available then a homeokinetically controlled robot can use this information more efficiently. This follows from the strongly enhanced sensitivity of the learning system and establishes a novel approach to learning in robots.

The homeokinetic principle suggests to use the so-called *time-loop error* (TLE) which is based on the reconstructed sensor values  $\hat{x}_t$ . Using Eq. 1 and assuming for now that  $\psi$  is invertible we define

$$\hat{x}_t = \psi^{-1}(\psi(x_t) + \xi_{t+1}) = \psi^{-1}(x_{t+1}) \quad (5)$$

which are sensor values that would have made the prediction perfect. Intuitively  $\hat{x}_t$  is obtained by going forward in time from  $x_t$  to  $x_{t+1}$  and then backward in time to  $\hat{x}_t$ . This sequence is called the time loop and thus the TLE is

$$E_{TLE} = \|v_t\|^2 \quad \text{with} \quad v_t = x_t - \hat{x}_t \quad (6)$$

which minimizes the mismatch between true sensor values  $x_t$  and their reconstruction  $\hat{x}_t$ .

In linear approximation we obtain  $v_t \approx L_t^{-1} \xi_{t+1}$ , where the matrix  $L_t = \frac{\partial \psi(x_t)}{\partial x_t}$  is the Jacobian of  $\psi$  at time  $t$ . Note that  $v_t$  can only be calculated after  $x_{t+1}$  is available. We account for non-invertible  $L$  by using a regularized inverse. The TLE

$$E_{TLE} = \|v_t\|^2 \approx \xi_{t+1}^\top (L_t L_t^\top)^{-1} \xi_{t+1}, \quad (7)$$

minimizes the norm of  $v$  (Eq. 6) and accounts for the error  $\xi$  (Eq. 1) only as much as it is transformed by the inverse dynamics of the system. This reveals another important feature of this error quantity, namely to minimize the norm of the inverse Jacobian. This results in an increase of predominantly the small eigenvalues of  $L$ . Therefore, the controller performs a destabilization in time. This eliminates the trivial fixed points (in sensor space) and enables spontaneous symmetry breaking which shows in the robot e. g. as a transition



from rest to a directed movement. Nevertheless, the system does not start to behave chaotically or enters uncontrollable oscillations because the destabilization is limited by the non-linearity  $g(\cdot)$  (Eq. 2). Intuitively, homeokinesis can be understood as the drive towards non-trivial behaviors that are still predictable by the internal model. Since the internal model is simple, smooth behaviors are preferred. Fig. 1 illustrates how the homeokinetic controller is connected to a robot.

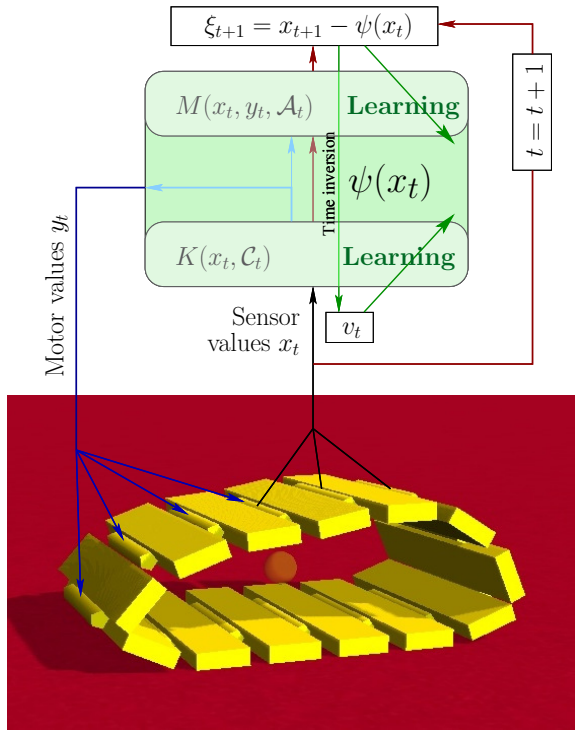


Figure 1: The homeokinetic controller connected to the ARMBAND robot. The ARMBAND consists here of  $m = 13$  flat segments that are connected by actuated joints. It receives sensory inputs  $x_i$  from the joint position sensors. The control architecture consists of the controller  $K$  and the predictor  $M$  which are combined to form  $\psi$ , see Eqs. 1 - 2. The transparent ball indicates the center of mass of the robot. It is used for evaluation of performance but not for control.

The TLE (Eq. 7) is minimized by gradient descent which gives rise to a parameter dynamics that evolves simultaneously with the state dynamics, see e. g. (Hesse et al., 2009).

$$\begin{aligned} C_{t+1} &= C_t - \epsilon_c \frac{\partial}{\partial C} E_{TLE} \\ h_{t+1} &= h_t - \epsilon_c \frac{\partial}{\partial h} E_{TLE} \end{aligned} \quad (8)$$

The learning rates  $\epsilon_c \approx \epsilon_A$  for the controller and the model are chosen such that the system adapts on the behavioral time scale. Because of unavoidable sensory noise, the TLE is never zero, neither does it have a vanishing gradient. The

rule (Eq. 8) produces therefore a continuously itinerant trajectory in the parameter space, i. e. the robot traverses a sequence of behaviors that are determined by the interaction with the environment. These behaviors are, however, waxing and waning and their transitions are hard to predict.

As an example, consider a robot with two wheels that is equipped with wheel velocity sensors. In the beginning the robot rests, but after a short time the homeokinetic learning rule initiates autonomous forward, backward or turning movements. If a wall is encountered that causes the wheels to stop, the robot will immediately reduce the motor speed and change the internal parameter to regain sensitivity. Eventually it will drive in a free direction. A more complex example for the self-organization of *natural* behaviors was provided by a spherical robot (Martius and Herrmann, 2010) that is actuated by movable internal masses. After a short time the robot starts to roll around one of its internal axes, but switches to a different axis every so often. Furthermore, high-dimensional systems such as serpentoid or catenoid robots, quadrupeds, hexapods and wheeled robots have been successfully controlled (see Martius et al. (2011)).

It is of particular interest that the control algorithm induces a preference for movements with a high degree of coordination among the various degrees of freedom. All the robotic implementations demonstrate the emergence of play-like behavior, which are characterized by coordinated whole body movements seemingly without a specific goal. The coordination among the various degrees of freedom arises from their physical coupling that is extracted and enhanced by the controller, because each motor neuron is adapted to be sensitive to coherent changes in all degrees of freedom due to Eq. 8.

## Guided Self-Organizing Control

How can we guide the joint dynamics of state (1) and parameters (8) in order to realize a given goal by the self-organizing process? One option is to modify the lifetime of the transient behaviors depending on a given reward signal, see Martius et al. (2007). A second and more stringent form of guidance was proposed by Martius and Herrmann (2010) and will be augmented and applied to a high-dimensional system in the present paper. We will formulate the problem in terms of problem-specific error functions (PSEF) that indicate an external goal by minimal values. A trivial example of such an error function is the difference between externally defined and actually executed motor actions. This is a standard control problem which, however, becomes difficult if the explorative dynamics is to be preserved.

GSO focuses on this interplay between the explorative dynamics implied by homeokinetic learning and the additional drives. The challenge in the combination of a self-organizing system with external goals becomes clear when recalling the characteristics of a self-organizing system. One important feature is the spontaneous breaking of symmetries

of the system. This is a prerequisite for spontaneous pattern formation and is usually achieved by self-amplification, i.e. small noisy perturbations cause the system to choose one of several symmetric options while the intrinsic dynamics then causes the system to settle into this asymmetric state. A nonlinear stabilization of the self-amplification forms another ingredient of self-organization. These two conditions which we will call our working regime, are to be met for a successful guidance of a self-organizing system. There are several ways to guide the homeokinetic controller which we will discuss in the following.

### Guidance by Problem-Specific Teaching

First we will describe how problem-specific error functions (PSEF) can be integrated. Recall that the adaptation of the controller parameters is done by performing a gradient descent on the time-loop error. The PSEF must depend functionally on the controller parameters in order to enable the same procedure. Unfortunately, the simple sum of both gradients (of the time-loop error and of the PSEF) is likely to steer the system out of its working regime. Furthermore, we cannot easily identify a fixed weighting between the two gradients that would satisfy an adequate pursuit of the goal while maintaining explorativity. One reason is that the non-linearity (Eq. 2) in the TLE causes the gradient to vary over orders of magnitude. A solution to this problem can be obtained by scaling the gradient of the PSEF according to the Jacobian matrix (see 7) of the sensorimotor loop such that both gradients become compatible. This transformation is essentially a natural gradient with the Jacobian matrix of the sensorimotor loop as a metrics. The update for the controller parameters  $C$  is now given by

$$\frac{1}{\epsilon_C} \Delta C_t = -\frac{\partial E_{TLE}}{\partial C} - \gamma \frac{\partial E_G}{\partial C} (L_t L_t^\top)^{-1}, \quad (9)$$

where  $E_G$  is the PSEF and  $\gamma \geq 0$  is the guidance factor deciding the strength of the guidance. For  $\gamma = 0$  there is no guidance and we re-obtain the unmodified dynamics (Eq. 8).

For clarity we will start with a very simple goal, namely we want a robot to follow predefined motor actions called *teaching signals* in addition to the homeokinetic behavior. We can define the PSEF as the mismatch  $\eta_t^G$  between motor teaching cues  $y_t^G$  and the actual motor values, thus

$$E_G = \|\eta_t^G\|^2 = \|y_t^G - y_t\|^2. \quad (10)$$

Since  $y_t$  is functionally dependent on the controller parameters (Eq. 2), the gradient descent can be performed, i.e. the derivative reads  $\frac{\partial E_G}{\partial C_{ij}} = -\eta_i^G g_i' x_j$ , where  $g_i' = \tanh' \left( \sum_{j=1}^n C_{ij} x_j + h_i \right)$  (all quantities at time  $t$ ). A similarly motivated approach is in linear systems is homeotaxis (Prokopenko et al., 2008).

An evaluation of the guidance mechanism has been performed using the TWOWHEELED robot, which was simulated in the realistic robot simulator LPZROBOTS (Martius

et al., 2011). The motor values determine the nominal wheel velocities and the sensor values report the actual wheel velocities of both wheels. We provided to both motors the same oscillating teaching signal. The resulting behavior is a mixture between the taught behavior and self-organized dynamics depending the value of  $\gamma$ . For  $\gamma = 0.01$  the teaching cues are followed most of the time but with occasional exploratory interruptions, especially when the teaching cues have a small absolute value. In this case the system is closer to the bifurcation point where the two stable fixed points for forward and backward motion meet. These interruptions cause the robot, for example, to move in curved fashion instead of strictly driving in a straight line as the teaching cue suggest. The exploration around the teaching signals might be useful in general to find modes which are better predictable or more active.

Interestingly, we can similarly define a mechanism that uses teaching cues in terms of sensor values (Martius and Herrmann, 2010).

### Guidance by Cross-Motor Teaching

Guidance mechanism can also use internal teaching signals. As an illustrative example, consider the mirror-symmetry that is preferred in many control systems. We will first follow this idea and describe a simple implementation following this example before we generalize this scheme later in order to apply it to high-dimensional systems. In either case, motor values of some motors will be used as teaching signals for other motors.

**Pairwise symmetries.** For two motors, guidance can be introduced by

$$y_{t,1}^G = y_{t,2} \quad \text{and} \quad y_{t,2}^G = y_{t,1}, \quad (11)$$

where  $y_t^G$  is the vector of nominal motor values, see (9, 10). For experimental evaluation we placed the TWOWHEELED robot in an environment cluttered with obstacles and performed many trials for different values of the guidance factor. The robot was rewarded for straight movement and was therefore expected not to get stuck at obstacles or in corners and cover substantial parts of its environment. In order to quantify the influence of the guidance we recorded the trajectory, the linear velocity, and the angular velocity of the robot. We expect an increase in linear velocity because the robot is to move straight instead of circling. For the same reason the angular velocity should be lowered. In Fig. 2 the behavioral quantification and a several sample trajectories are plotted. Additionally the relative area coverage is shown, which indicates that much more area of the environment was covered by the robot with guidance compared to freely moving robot. As expected, the robot shows a distinct decrease in mean turning velocity and a higher area coverage with increasing values of the guidance factor until the guidance becomes dominant and the performance drops. In the

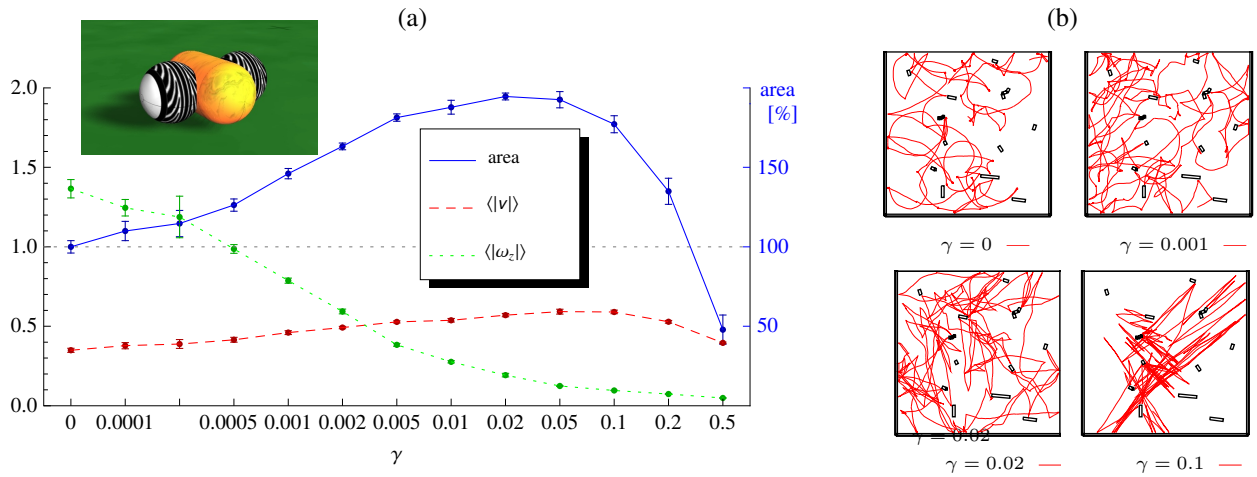


Figure 2: Behavior of the TWOWHEELED robot when guided to move preferably straight. **(a)** Mean and standard deviation (of five runs each 20 min) of the area coverage (**area**), the average velocity  $\langle |v| \rangle$ , and the average angular velocity  $\langle |\omega_z| \rangle$  for different values of the guidance factor  $\gamma$ . Area coverage (box counting method with  $300 \times 300$  boxes) is given in percent relative to case  $\gamma = 0$  (**right axis**). The robot is driving straighter and its trajectory covers more area for larger  $\gamma$ . The inset shows a screenshot of the simulation. **(b)** Example trajectories for different guidance factors. Note that for  $\gamma = 0.1$  still many boxes are visited but less well spread. Parameters:  $\epsilon_c = \epsilon_a = 0.01$ , update rate 100 Hz.

normal regime the robot is still performing turns and drives both backwards and forwards and that it does not get stuck at the walls, as seen in the trajectory in Fig. 2(b), is because the sensitivity (exploration) and predictability (exploitation) of the controller remain. If the guidance is too strong the favorable properties of the self-organizing behavior are lost such that the robot stalls or performs repetitively the same motion. Note that already very small values of  $\gamma$  yield a high effect of the guidance.

**Permutation relations.** In a more general cross-motor teaching setup, each motor has one incoming and one outgoing connection, such that there is still only one teaching signal per motor. The connections can be described by a permutation  $\pi_m$  of  $m$  motors that assigns each motor a source of teaching input. The teaching signal is then given by

$$(y_t^S)_i = (y_t)_{\pi_m(i)} \quad \text{for } i = 1, \dots, m. \quad (12)$$

With a cyclic schema of connections a group of motors can be synchronized. In the following experiment we use a rotation-symmetric motor connection setting to show that a high-dimensional chain-like robot can quickly develop a locomotion behavior.

The ARMBAND robot consists of a sequence of flat segments placed in a ring-like configuration, where subsequent segments are connected by motor-operated hinge joints. As a result we obtain a robot with the appearance of a bracelet or chain, see Fig. 1. Each joint provides a sensor value of the current position. The motor values define target joint positions, which typically cannot be reached due to substantial

physical constraints and underactuation. In this way the controller obtains informative feedback from the robotic body. Since the robot is symmetric there is by construction no preferred direction of motion, meaning that the homeokinetically controller robot will move forward or backward with equal probability. The robot cannot turn or move sideways, but it can produce a variety of postures and locomotion patterns.

With the method of cross-motor teaching we can select different symmetries, such that the robot is more likely to perform a directed motion. For that we define the permutation used in Eq. 12 as

$$\pi_m(i) = (i + k + \lfloor m/2 \rfloor) \mod m, \quad (13)$$

where  $k \in \{-1, 0, 1\}$ . Coarsely speaking, this connects motors on the opposite side of the robot with a shift to one or the other side in a way that depends on  $k$ . The choice of  $k$  reflects the desired direction of motion and depends on whether the number of joints  $m$  is even or odd. If  $m$  is even then  $k = -1$  and  $k = 1$  are used for both directions (forward or backward) and  $k = 0$  represents a point symmetric connection setup. In the latter case the robot will not prefer a direction of motion and the behavior is similar to the case without guidance. For odd values of  $m$ , which is used here,  $k = 0$  and  $k = 1$  need to be used, resp., for backward and forward motion. In the following experiments the robot has  $m = 13$  motors. The motor connections for  $k = 1$  are shown in Fig. 3. Each motor connection is displayed by an arrow pointing to the receiving motor. Note that the connections are directed and a motor is not teaching the same motor from which it is receiving teaching cues. For  $k = 0$  (and  $n$

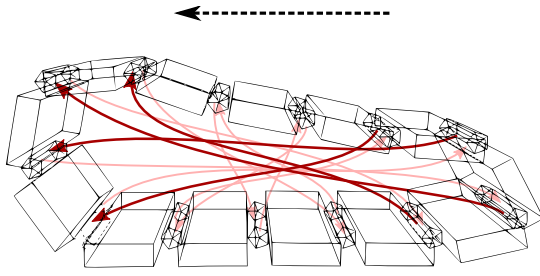


Figure 3: ARMBAND robot with cross-motor connections. Links are connected by hinge joints that are actuated by servo motors. The curved arrows indicate unidirectional cross-motor connections. For these connections the robot preferably moves leftwards. All links are identical, but four links are drawn boldly for better visibility.

odd) all arrows are inverted, meaning that for each connection the sending and receiving motors would swap roles.

## Results

To evaluate the performance we conducted for different values of the guidance factor  $\gamma_S$  five trials each 30 min long. In a first setting the cross-motor connections were fixed ( $k = 1$ ) for the entire duration of the experiment. We observed the formation of a locomotion behavior after a very short time. Note that this behavior requires all joints of the robot to be highly coordinated. As a quantitative measure of the performance we calculate the horizontal velocity  $v$  using the center of mass of the robot. Thus, the velocity is a scalar and we define forward motion if  $v > 0$  and backward motion if  $v < 0$ . In this experiment we expect the robot to move only forward, because a fixed cross-motor connection setup was used. The average velocity of the robot increased distinctively with raising guidance factors, see Fig. 4(a). For excessively large values of the guidance factor  $\gamma_S$  the velocity goes down again. This occurs for two reasons: First, the cross-motor teaching has a too strong influence on the working regime of the homeokinetic controller and second the actual motor pattern of the locomotion behavior does not perfectly obey the relations between the motor values, not all motor values are exactly equal. Again, already a small value of  $\gamma$  is sufficient to achieve the goal. It appears the self-organizing system needs only very little influence to be guided into the desired regions of the behavior space.

Without guidance the robot moves equally to both directions but with comparably low velocity. This can be seen at the mean of the absolute velocity in Fig. 4(a). If the value of the guidance factor is chosen conveniently, the robot moves in one direction with varying speed see Fig. 4(b) for 3 velocity traces. The velocity traces are seen to have a peak followed by a dip before a more steady regime is attained. It appears that the controller learning surpasses a more optimal

configuration with respect to the velocity, but there the trade-off between self-organizing and guidance is not met. Later strong fluctuations may occur that reflect the explorative nature of the homeokinetic part. The locomotive behavior can also be seen in Video 1, see Ref. (Supplement, 2011), for a low value of guidance factor ( $\gamma_s = 0.001$ ) and in Video 2 for a medium value of guidance factor ( $\gamma_s = 0.003$ ).

In a second setup, we changed the cross-motor connections every 5 min, i. e.  $k$  was changed from 0 to 1 and back. A value of  $k = 0$  should lead to a negative velocity and a  $k = 1$  to a positive velocity. To study the dependence on the guidance factor and to measure the performance we use the average absolute velocity ( $\langle |v| \rangle$ ) and the correlation of the velocity with the configuration of the connections ( $\rho(v, k)$ ), see Fig. 5(a). Without guidance ( $\gamma_S = 0$ ) there is, as expected, no correlation with the supposed direction of locomotion. For a range of values of the guidance factor we find a high total locomotion speed with a strong correlation to the supposed direction of motion. Note that the size of the correlation depends on the length of the intervals of one connection setting. For long intervals the correlation will approach one. In Fig. 5(b) the velocity of the robot is plotted for different runs with the same value of the guidance factor that was used in the previous experiment ( $\gamma_S = 0.003$ ). We observe that the robot changes the direction of motion shortly after the configuration of connections was changed, see also Video 3 at Supplement (2011).

The locomotion of the robot is essentially influenced by the number of cross-motor connections. For that we use again the fixed connectivity. In a series of simulations a number  $0 \leq l \leq m$  equally spaced cross-motor connections (Fig. 3) are used. With increasing  $l$  the robot start to locomote earlier. Full performance is reached already if 8 out of the 13 connections are used, see Fig. 6(a).

In order to study the scaling properties of the learning algorithm we varied the number of segments  $m$  of the robot and thus the dimensionality of the control problem. The results are astonishing, see Fig. 6(b): The behavior is learned with the same speed also for large number (40) of segments. There is no scaling problem here for the following reason. In the closed loop with an approximate feedback strength (self-regulated by the homeokinetic controller) the robot needs only very little influence to roll. The length of the robot can even help because other behavioral modes (e. g. wobbling) are damped increasingly due to gravitational forces. For the same reason, small robots are slower than medium ones. Large robots are again slower because the available forces at the joints become too weak. The experiment illustrates that specific behaviors can be achieved in a high-dimensional robot by using cross-motor teachings. Cross-motor connections can break the symmetry between the two directions of motion such that a locomotory behavior is produced quickly. When the connections are switched later during runtime, the behavior of the robot changes reliably.

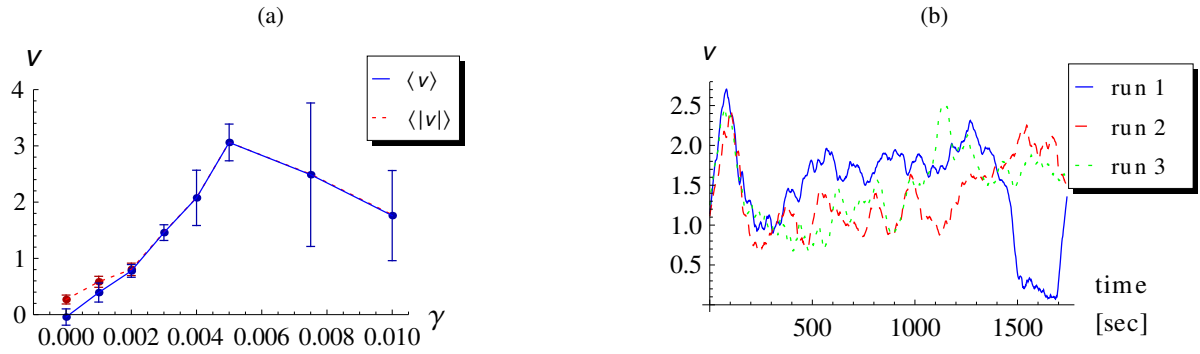


Figure 4: Performance of the ARMBAND robot with constant cross-motor teaching. **(a)** Mean and standard deviation of the average velocity  $\langle v \rangle$  and the average absolute velocity  $\langle |v| \rangle$  of five runs for different guidance factors  $\gamma_S$ . **(b)** Velocity of the robot  $\bar{v}$  (average over 1-minute sliding window) for three runs at  $\gamma_S = 0.003$ ,  $k = 1$ ,  $\epsilon_c = \epsilon_a = 0.1$ , 100 Hz update rate.

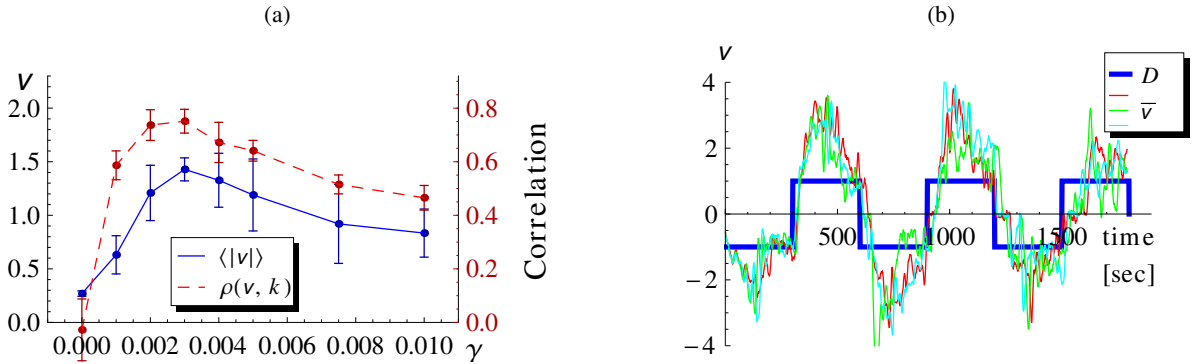


Figure 5: Performance with switching cross-motor teaching. **(a)** Mean and standard deviation of the average absolute velocity  $\langle |v| \rangle$  and the correlation  $\rho(v, k)$  of the velocity with the configuration of the connections of five runs for different guidance factors  $\gamma_S$ . **(b)** Velocity (average over 10-seconds sliding window) for three runs of the robot with a supposed direction of motion  $D$ . Parameters as in Fig. 4.

The guidance mechanism can also be transferred to sensor space using the direct sensor teaching, which was discussed above and was proposed by Martius and Herrmann (2010). One obtains a cross-sensor teaching analogously to the definitions given above. This can become useful, for example, if a certain behavior is demonstrated by a human operator by passively moving the robot. In the case of the ARMBAND robot, one can easily imagine that the robot is pushed along the ground such that a locomotion pattern is formed. Based on the sensor readings, the correlations between the sensor channels can be determined and serve as a basis for the construction of a specific cross-sensor teaching configuration.

## Discussion

We have presented here two mechanisms to guide the homeokinetic self-organization of behavior. The first one uses desired motor patterns that were introduced into the learning dynamics by means of an additional error function. The strength of guidance can be conveniently adjusted. We have considered also cross-motor teaching as a new way of using the directed teaching to select desired behaviors. The

approach introduced here is realized by a permutation of the motors signal for teaching. We applied this algorithm to a bracelet-like robot (ARMBAND) with many degrees of freedom and demonstrated the accelerated development of locomotion behavior from scratch. Even the relearning to the opposite direction of motion is possible very quickly. Since the learning is very fast and the performance changes gradually with changing  $\gamma$ , the guidance factor could be adapted automatically. Most striking is the scaling of the algorithm to higher dimensions. In the present case the performance did not decrease when the robot was enlarged to have 40 DoF. This is a result of the exploitation of the embodiment by the self-organization process.

The exploratory character of the controller is retained under guidance and helps to find a behavioral mode even if the specification of the motor teaching signals are partially contradictory. For example, the TWOWHEELED robot can choose freely between driving forward or backward, because the behavior-space is only partially constrained. Furthermore, it is evident that the robot remains sensitive to small perturbations and continues to explore its environment. The



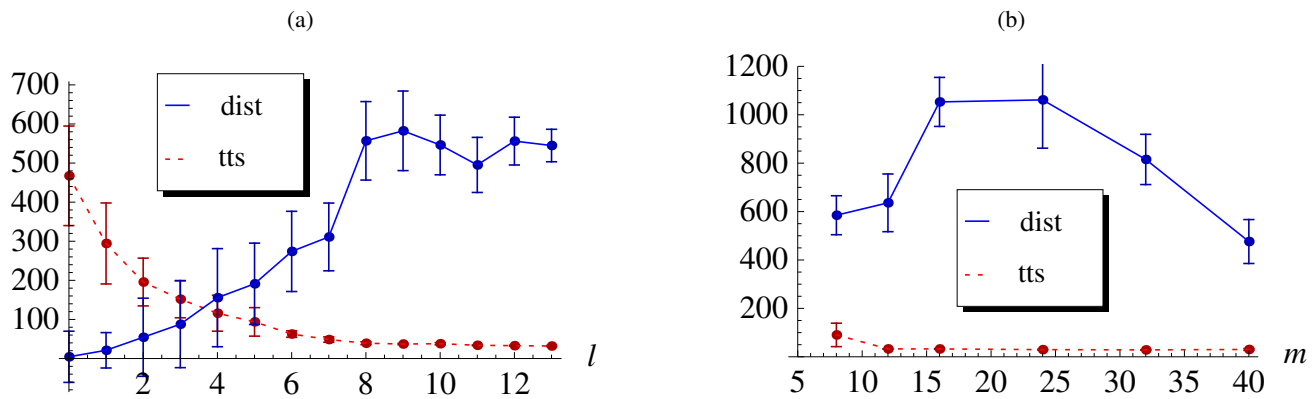


Figure 6: Scaling of learning time and performance for different robot complexity. The plots show mean and standard deviation of the distance traveled by the robot ('dist' in units of 1 segment size) and of the time-to-start ('tts' in seconds) of 20 runs à 10 min ( $\gamma = 0.003$ ). (a) Performance as a function of the number of cross-motor connections  $l$  (equally spaced around a robot with  $m = 13$  joints). (b) Performance for different numbers of segments  $m$  (DoF) with full cross-motor connectivity ( $l = m$ ).

constraints are not strictly enforced by the algorithm but the self-organization can find a mode that fits better to the particular embodiment. The presented experiments with the ARMBAND demonstrate this effect. The guidance signal alone would synchronize all motors to the same value (same phase in the oscillations) which does not lead to a locomotion behavior whereas the combined learning dynamics leads to a smooth and adaptive locomotion, see Video 3 (Supplement, 2011).

**Acknowledgment:** The project was supported within the National Bernstein Network by the BMBF grant #01GQ0432 at BCCN Göttingen. We are grateful to Ralf Der and Theo Geisel for encouragement and discussions.

## References

- Der, R. (2001). Self-organized acquisition of situated behaviors. *Theory in Biosciences*, 120:179–187.
- Der, R., Hesse, F., and Martius, G. (2006). Rocking stamper and jumping snake from a dynamical system approach to artificial life. *Adaptive Behavior*, 14(2):105–115.
- Herrmann, J. M. (2001). Dynamical systems for predictive control of autonomous robots. *Theory in Biosciences*, 120:241–252.
- Hesse, F., Martius, G., Der, R., and Herrmann, J. M. (2009). A sensor-based learning algorithm for the self-organization of robot behavior. *Algorithms*, 2(1):398–409.
- Jung, T., Polani, D., and Stone, P. (2011). Empowerment for continuous agent-environment systems. *Adapt. Beh.*, 19:16–39.
- Lopez, M. and Oudeyer, P.-Y. (2010). Active learning and intrinsically motivated exploration in robots: Advances and challenges. *IEEE Transactions on Autonomous Mental Development*, 2(2):65–69.
- Martius, G., Fiedler, K., and Herrmann, J. M. (2008). Structure from behavior in autonomous agents. In *Proc. IEEE IROS 2008*, pages 858 – 862.
- Martius, G. and Herrmann, J. (2010). Taming the beast: Guided self-organization of behavior in autonomous robots. In Doncieux, S. et al., editors, *From Animals to Animats 11*, volume 6226 of *LNCS*, pages 50–61. Springer.
- Martius, G., Herrmann, J. M., and Der, R. (2007). Guided self-organisation for autonomous robot development. In Almeida e Costa, F. et al., editors, *Proc. Advances in Artificial Life, (ECAL 2007)*, volume 4648 of *LNCS*, pages 766–775. Springer.
- Martius, G., Hesse, F., Güttler, F., and Der, R. (2011). LPZROBOTS: A free and powerful robot simulator. <http://robot.informatik.uni-leipzig.de>.
- Nolfi, S. (2006). Behaviour as a complex adaptive system: On the role of self-organization in the development of individual and collective behaviour. *ComplexUs*, 2(3-4):195–203.
- Oudeyer, P.-Y., Kaplan, F., Hafner, V. V., and Whyte, A. (2005). The playground experiment: Task-independent development of a curious robot. In *AAAI Spring Symp. on Developmental Robotics, 2005*, pages 42–47, Stanford, California.
- Prokopenko, M. (2009). Guided self-organization. *HFSP Journal*, 3(5):287–289.
- Prokopenko, M., Zeman, A., and Li, R. (2008). Homeotaxis: Coordination with persistent time-loops. In Asada, M. et al., editors, *SAB*, volume 5040 of *LNCS*, pages 403–414. Springer.
- Schmidhuber, J. (1991). A possibility for implementing curiosity and boredom in model-building neural controllers. In Meyer, J. A. and Wilson, S. W., editors, *Proc. Simulation of Adaptive Behavior: From Animals to Animats*, pages 222–227. MIT.
- Singh, S., Barto, A., and Chentanez, N. (2004). Intrinsically motivated reinforcement learning. In *Proc. 18th Annual Conf. Neural Information Proc. Systems*, Vancouver, BC, Canada.
- Supplement (2011). Videos for this article. <http://robot.informatik.uni-leipzig.de/research/supplementary/ECAL2011>.
- Tani, J. (2003). Learning to generate articulated behavior through the bottom-up and the top-down interaction processes. *Neural Networks*, 16(1):11–23.

# A Robotic Approach to Understand the Role of Vicarious Trial-and-Error in a T-Maze Task

Eiko Matsuda<sup>1</sup>, Julien Hubert<sup>1</sup> and Takashi Ikegami<sup>1</sup>

<sup>1</sup>The university of Tokyo, 3-8-1, Komaba, Meguro-ku, Tokyo, 153-8902, Japan  
{eiko,jhubert,ikeg}@sacral.c.u-tokyo.ac.jp

## Abstract

Vicarious trial-and-error(VTE) is a type of conflict-like behavior, observed in route selection tasks (Tolman (1939)). Studies of VTE have shown a correlation between the number of VTEs exhibited by a system with its learning efficiency. At the onset of learning a task, the number of VTEs increases, and when the learning reaches its plateau, it decreases.

The question we explore in this paper concerns the role of VTE. Basing ourselves on a model developed by Bovet and Pfeifer (2005), we ran robotic experiments to compute the number of VTEs during the learning of a T-maze task. Our results first show that what has been found in rats can be replicated in artificial systems. Furthermore, by changing the connectivity pattern of the original model, we discovered that the connection between VTEs and learning efficiency might not be necessarily true as our results show that two models exhibiting the same performance can possess a different pattern of VTEs. By comparing the robustness of the two models under varied conditions, we propose that VTEs are connected to the adaptivity of a system to environmental changes.

## Introduction

In his experiments, Tolman (1939) observed that rats are seemingly hesitating when they must choose between one of two rooms, one of which containing a reward while the other being empty. The only cue differentiating the rooms is the color of their doors. A black door indicates the room provides a reward, and a white color indicates an empty room. To reach the reward, the rats must learn the relationship between the color of the door and the presence of the reward. During the learning phase, the rats have been seen moving their head from one door to another which is referred by Tolman as a conflict-like behavior named "vicarious trial-and-error (VTE)". In his experiments, Tolman noticed that the number of VTEs increases at the onset of the learning phase to start decreasing when the performance reaches its plateau. From that observation, VTE has been connected to learning efficiency.

Following Tolman's observations, other researchers started paying attention to the presence of VTEs in their studies. Hu and Amsel (1995) showed hippocampal contribution to VTEs. Johnson and Redish (2007) reported the

presence of VTEs in experiments on rats who were shown to be simulating their next decisions internally before acting. Tarsitano (2006) found that, in a detour task, jumping spiders display two phases of action: the inspection phase, where spiders stop and inspect possible routes toward a target, and the locomotory phase, where spiders move toward a single direction. VTEs have been observed during the inspection phase. Tarsitano concluded that "one can speculate that it is a small but significant jump to use trial and error vicariously when choosing a goal to approach". Ikegami (2007) suggested the relationship between VTEs and private simulation. From these researches, VTE seems to have some essential role in internal reflection and decision making. However, the role of the VTEs has yet to be fully investigated.

The question we explore in this paper concerns the role of VTE. Using a model developed by Bovet and Pfeifer (2005) for T-maze learning experiments on robotic platforms, we study the presence of VTEs during the acquisition of the task. Our results display the same pattern of increase followed by a decrease in the number of VTEs as observed in the rat. Additionally we vary environmental parameters as well as the connectivity of the network in order to study the variations in the number of VTEs. Based on our results we hypothesize that VTEs might be connected to robustness and adaptivity. We first detail the environmental setup and the neural model in the next two sections. Then the results will be presented with a discussion of their significance.

## Methodology

Our work is based on a robotic and neural model developed by Bovet and Pfeifer. The model combines five types of modalities to control a robot in a T-maze task. The neural model is self-organized with no hierarchy between modalities, nor predetermined sensori-motor relationship. Modalities are associated through Hebbian learning only (Hebb (1949)).



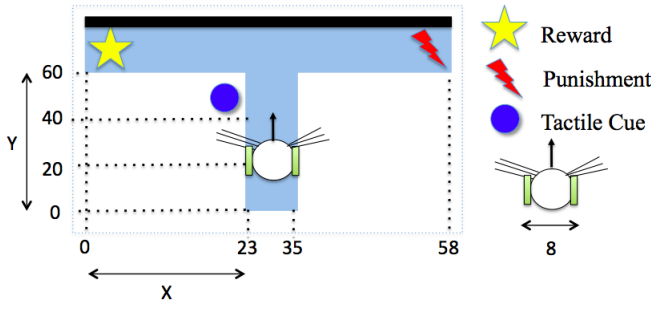


Figure 1: T-maze environment used for the experiment. At the beginning of each trial, the robot is placed on the central arm of the maze. The circle at the choice point represents the tactile cue, the star at one end of the maze indicates reward, and the lightning at the other end of the maze stands for punishment. The back wall is painted black and the other walls are white, which are detected by the robot's omnidirectional camera. Walls of the T-maze are perceived by the robot's proximity sensors. The length and the width of the T-maze are denoted by 'X' and 'Y'.

### Experimental setup

The environment is a T-maze with one central arm and two side ones (see figure 1). A reward is located at the end of one arm, and a punishment is placed on the opposite one. The robot learns to reach the reward following a tactile cue placed at the end of the central arm, on the same side as the reward.

The robot is modeled following the e-puck robot (Mondada et al. (2009)) and is equipped with the following sensors and motors:

- 1) **Tactile sensors:** Tactile stimulation comes from 32 whiskers attached to the left and right sides of the robot. The signal is binary, on or off. Whisker sensors detect the tactile cue at the intersection point of the T-maze. The walls of the T-maze are low enough so that the whiskers can only detect the cue.
- 2) **Vision sensors:** Visual stimulation reflects the activity of the omnidirectional camera, which return grayscale values standardized from 0 to 1. This camera is composed of 20 pixels aligned horizontally. Everything in the T-maze is made white or transparent, except for the black back wall. In other words, the omnidirectional camera gets positive signals only from the black wall at the back of the T-maze. By this modality, the robot acquires destination information.
- 3) **Proximity sensors:** Six proximity sensors are regularly attached to the front half of the body. These sensors detect the distance from the robot to the walls of the T-maze. The values are standardized between 0 and 1. These sensors

are attached low enough to only detect the walls of the T-maze. This modality is involved in the experiment only indirectly to achieve wall avoidance.

- 4) **Reward sensitivity:** The reward sensitivity is usually set to 0. It is raised to 1 to signal a reward and lowered to -1 to indicate punishment. The value is dependent on which side of the maze is reached by the robot.

- 5) **Motors:** The forward velocity of the robot  $v_f$  is constant and positive, and the turning degree  $v_t$  is determined by the neural controller of the robot which is explained later. Both  $v_t$  and  $v_f$  are standardized between 0 and 1, and activate actual left and right wheel velocities,  $v_l, v_r$ , as following:

$$\begin{pmatrix} v_l \\ v_r \end{pmatrix} = C \cdot \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} v_f \\ v_t \end{pmatrix} \quad (1)$$

where  $C$  is a constant for converting the standardized value to the actual motor speed. If  $v_t > 0$ , then  $v_l < v_r$ , which makes the robot turn left, and  $v_t < 0$  produces a right turn.

As a training phase, the robot runs randomly in an empty maze with no tactile cue nor reward signals. Afterward, the tactile cue and the reward are introduced into the T-maze, and the robot must complete the task. The robot learns the correlation between modalities through Hebbian learning while acquiring a reward seeking behavior (for more detailed explanations, see Bovet and Pfeifer (2005)).

### Neural Network - Bovet et al.'s Original Model

The neural network is composed of five modality modules: tactile, vision, proximity, reward, and motor (figure 2(a)). Each of them plays a separate role in treating the signals from its corresponding sensor (or motor) on the robot. Each modality has five types of neural populations, described in figure 2(b). These five populations are composed of the same number of artificial neurons, this number varying depending on the type of modality. For instance, the tactile modality has 32 neurons for each of the five populations while the motor modality has only one neuron per population. The five types of populations are described as follows:

- 1) **Current state** The current state of modality  $M$ ,  $\mathbf{x}^M(t) = (x_1^M(t), x_2^M(t), \dots, x_m^M(t))$ , receives signals from the corresponding sensors (or motors). For instance, tactile stimuli from 32 whisker sensors activate the corresponding 32 nodes of the current state.
- 2) **Delayed state** The delayed state  $\tilde{\mathbf{x}}^M(t) = \mathbf{x}^M(t - \tau)$  receives signals  $\tau$  timestep in the past.
- 3) **Current state change** The current state change  $\mathbf{y}^M(t)$  is the difference between the current and the delayed state,

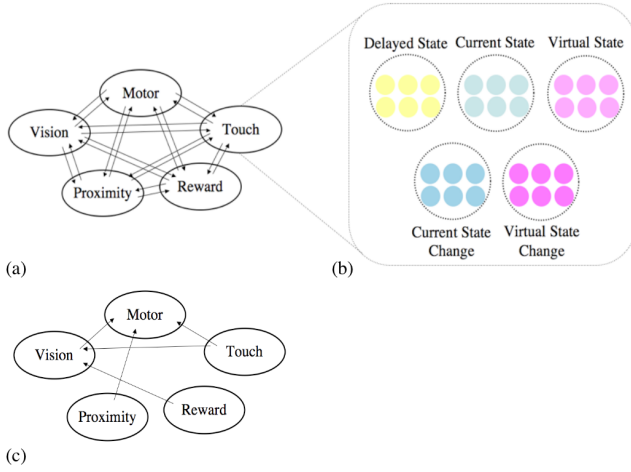


Figure 2: (a) Five sub-systems for each modalities make up the whole cognitive system of the robots. These modalities are fully connected with each other. (The original model) (b) Five types of neural populations, called the current state, the delayed state, the current state change, the virtual state, the virtual state change. (c) A new neural network where the five modalities are sparsely connected sparsely. (The minimal model)

as described below:

$$\begin{aligned} \mathbf{y}^M(t) &:= \mathbf{x}^M(t) - \tilde{\mathbf{x}}^M(t) \\ &= \mathbf{x}^M(t) - \mathbf{x}^M(t - \tau) \end{aligned} \quad (2)$$

**4) Virtual state** The virtual state of modality  $M$ ,  $\tilde{\mathbf{x}}^M(t)$ , is activated by the virtual state change of other modalities:

$$\tilde{\mathbf{x}}^M(t+1) := f(\sum_{N \neq M} W^{MN}(t) \cdot \tilde{\mathbf{y}}^N(t)) \quad (3)$$

where  $W^{MN}$  is the weight matrix connecting modality  $M$  to modality  $N$  and  $f(x)$  is a sigmoid function, written as:

$$f(x) = \frac{1.0}{1.0 + \exp(-a \cdot x)} \quad (4)$$

**5) Virtual state change** The virtual state change of modality  $M$ ,  $\tilde{\mathbf{y}}^M(t)$ , is the difference between the virtual and the current state.

$$\tilde{\mathbf{y}}^M(t) := \tilde{\mathbf{x}}^M(t) - \mathbf{x}^M(t) \quad (5)$$

The current state population, delayed state population and current state change population do not possess any in or out connections toward other modalities. All virtual state change populations are connected to virtual state populations of the other modalities. For instance, the neurons from the virtual state change population of the tactile modality

are connected to the neurons of the virtual state population of all the other modalities. Those connections are the only ones present in the model.

All the connections of the model are tuned using a modified version of Hebbian learning. The main difference between Hebb's version is that the pre and post synaptic neurons are not used to compute the change of the synaptic connections. For the learning, the neurons of the non-virtual populations are used. Instead of the virtual state population, the neurons of the current state population are used for the Hebbian learning. Similarly, the neurons of the current state change population replace the ones from the virtual state change population. Mathematically, this corresponds to the following equations:

$$\begin{aligned} \Delta W^{MN}(t) &:= l \cdot (\mathbf{x}^N(t) \mathbf{y}^M(t)^T - \alpha \mid \mathbf{y}^M(t) \mid W^{MN}(t)) \\ W^{MN}(t+1) &= W^{MN}(t) + \Delta W^{MN}(t) \end{aligned} \quad (6)$$

where  $l$  is the learning rate and  $\alpha$  is the forgetting rate. Because of the  $\alpha$ , the weight between a pair of neurons is decreased if the two neurons are not activated at the same time. It also prevents the weights from growing to infinity.

To clarify the inner algorithm of the neural model, we detail the steps leading to the generation of the outputs as follows:

1. Sensory information is transferred to the current state population.
2. The delayed state population is updated, followed by the current state change population.
3. Hebbian learning is applied on all the connections of the model.
4. The activity of the neurons from the virtual state change populations of all modalities are propagated to the neurons of the virtual state populations using a feedforward algorithm (see equation 3).
5. The activation of the single neuron of the virtual state population from the motor modality is assigned to the output  $v_t$  from equation 1.

For additional details on this model, please refer to the original paper Bovet and Pfeifer (2005).

## Neural network - Minimal model

In addition to the original neural network invented by Bovet and Pfeifer, we conducted experiments with a new neural network model. In the original model, modalities are fully connected (figure 2(a)), while our new model has only minimal connectivity among modalities. By "minimal" connectivity, we mean connections which have a specific role to solve the task. Those connections are mentioned in Bovet's

paper and stem from his analysis of the neural network, as shown in figure 2(c). We expect that the behavioral difference between the original and the minimal connectivity model will allow us to uncover the role of VTE.

### Setup of the Genetic Algorithm

Bovet and Pfeifer’s model relies on the following parameters: learning rates and forgetting rates for each modalities, update frequency of the neural network,  $\tau$  for the delayed state and constants for the sigmoid function in equation 4. Despite the authors not mentioning how to select those parameters, we found out that slight differences in their value can strongly influence the performance of the robot. This is partly due to our experiments adopting more tolerant conditions than the original experiment, like a broader T-maze. To tune these parameters and optimize the performance of the controller, we employ a genetic algorithm (GA)(Holland (1975)). Our GA possesses a population of 100 individuals to optimize 59 parameters using tournament selection, single point crossover applied with a probability of 70% and a 1% mutation rate. We also use elitism by simply copying the 5 best individuals directly to the next generation. A fitness function  $F(t)$  at generation  $t$  is calculated as:

$$F(t) = \begin{cases} +5 \text{ points, if it reaches the reward.} \\ +0.25 \text{ points, if it reaches the punishment.} \\ +0 \text{ points, if it gets timeout.} \end{cases} \quad (7)$$

The amount of points assigned is determined arbitrarily. The trials are repeated 100 times from one fixed initial position, which gives a maximum fitness value of 500. We conducted several runs of the GA for the original and the minimal model respectively.

### Results

For each model, we evolved 5 runs of GA. Figure 3 shows 2 out of 5 GA runs get the maximum fitness value (100% success) with the original model, and, in the case of the minimal model, 3 out of 5 GA runs successfully evolved. We selected one individual for each model from these evolved runs and counted the number of VTE they displayed.

It is important to notice that evolution produces different strategies for both models and that not all of them shows VTEs. For that reason we chose to work on 5 runs of GA but more have been evolved and analyzed despite not being presented here.

Our methodology to count the number of VTE in a robot is similar to the one used by Tolman. In our case, the robot does not possess a head moving independently from its body so the whole body movement has to be considered. One VTE is granted if the turning degree  $v_t$  from the equation 1 changes its sign. In order to filter small oscillations around a turning degree of 0, a VTE is only granted if the sign change is outside the range  $[-0.3; 0.3]$ .

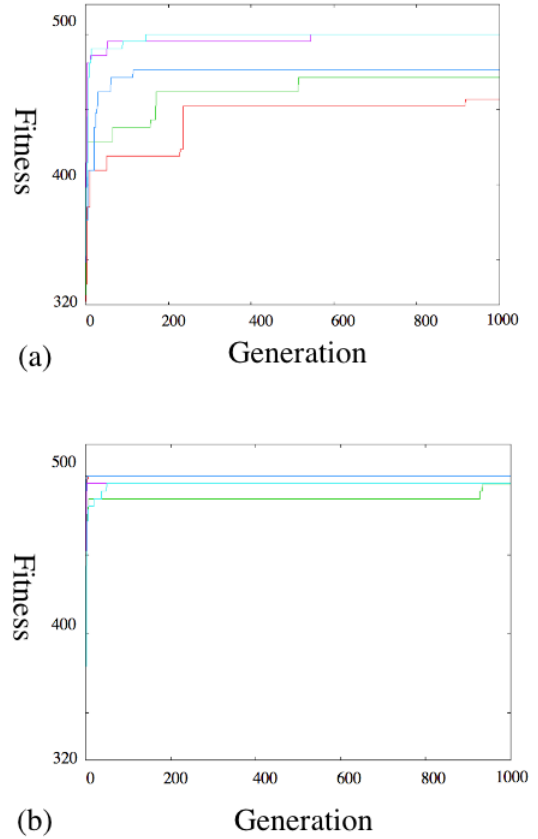


Figure 3: Fitness values of the five runs of GA. The X axis represents the number of generations, and the Y axis the fitness of the best individual in each generation. Maximum fitness value is 500. (a) In the case of the original model. Two out of five runs of GA get the maximum value. (b) In the case of the minimal model. Three out of five runs get the maximum value.

Figure 4 shows the number of VTEs observed for one evolved individual for each model. We can see from figure 4(a) that the robot evolved with the original connectivity model exhibits more VTEs at the beginning of the learning, to decrease afterward. This observation is similar to Tolman’s experiments on real rats (Tolman (1939); Muenzinger and Fletcher (1934)). On the other hand, the robot with the minimal connectivity model shows VTEs in a lower amount while remaining constant during the course of the experiment (figure 4(b)). Despite this difference in the number of VTEs, both models show a success rate of 100%. This result implies that VTEs are not directly related to performance in learning, but might have another purpose.

In order to try to understand what differentiates the two models, we looked at how the synaptic strengths are chang-

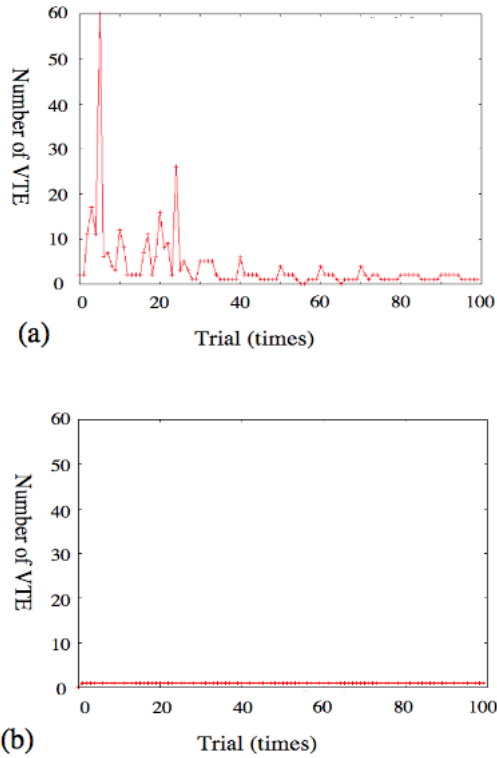


Figure 4: Change in the number of VTE during learning. (a) In the case of the original model. (b) In the case of the minimal model.

ing during 100 trials. Figure 5 presents the variation of the weights from the minimal model with their equivalent from the fully connected one. If the weights responsible for the VTEs were present among those, we would expect the strength to vary initially to stabilize toward the end of the trial as a similar observation was done on the VTEs. In the minimal connection model, no weights show such a pattern of variation. The strengths of the weights remain periodic over all the trials. In the case of the fully connected model, proximity(IR) to motor, touch to vision and vision to motor display non periodic variations, oscillating initially to stabilize later on. Proximity to motor is even more interesting as it decreases progressively over all the trials. Those variations show that the robot is changing its behavior progressively over all the trials. In the case of the minimal model, the robot does not seem to modify its strategy to reach the goal as the weights seem to be periodic. This analysis alone does not explain the source of the VTEs but it implies that the VTEs are not a random behavior and might be caused by the dynamics of the neural network.

In order to study if the presence of VTEs could imply a higher level of robustness for the robot, we analyze the

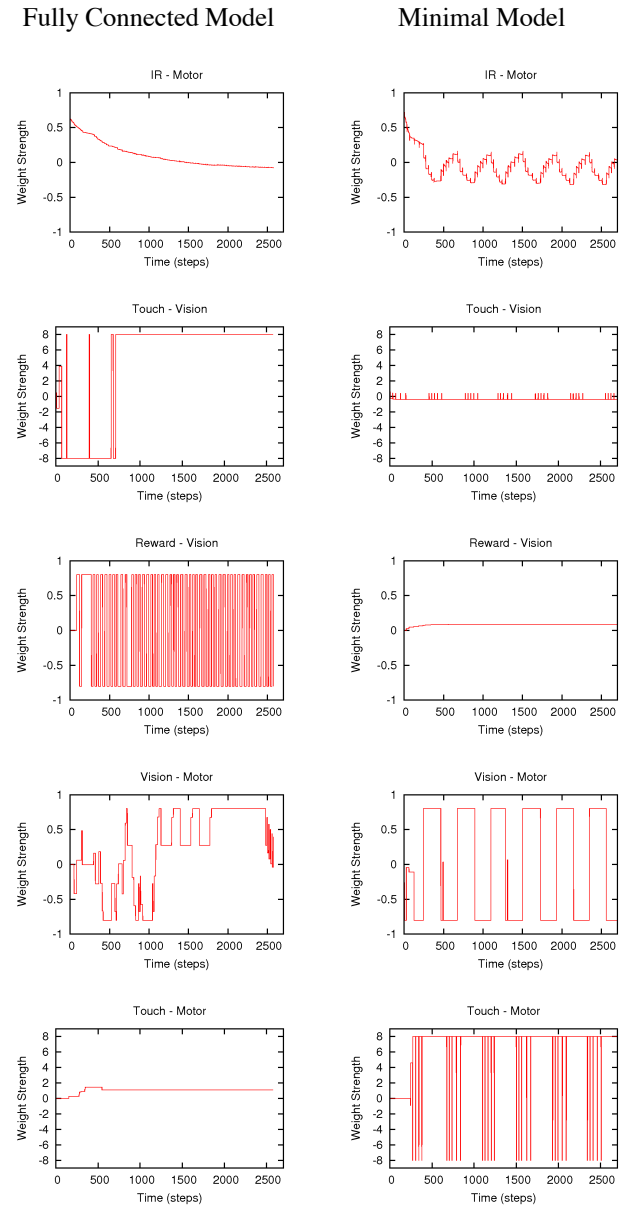


Figure 5: Comparison between the variations of the weights from the minimal model with their equivalent in the full model over 100 trials.

performance under varying initial conditions. During evolution, the starting position is  $(x, y) = (29, 20)$ . This experiment explores if the performance of the robot is affected by a change in its initial position by testing it from every other starting position inside the central arm of the T-maze. Each position has been tested 100 times to obtain the final results.

Figure 6 shows the results for each model. The first observation from this figure is that the performance is not constant over all starting positions. Some areas lead to higher performances. The second observation concerns the comparison of the variance of the performance between the two models.

In the case of the original model, the variance remains under 400 while the minimum variance of the minimal model is around 600 as seen in figure 7. This means that, despite the two models having a similar average performance, the original model seems to withstand changes in starting position. On the other hand, the minimal model is strongly affected by the initial position. This result implies that the presence of VTEs could be associated with a higher level of robustness to changes in the environment.

Based on the success rate, we observed 5 different types of behaviors:

**Going to the reward** As we described above, the robot successfully reaches the reward side. In the case of the original model, the number of VTEs becomes higher at the beginning of the learning, and decrease afterward similarly to experiments with real rats (figure 4). But with the minimal model, we only observed lower and stable VTEs.

**Going to the punishment** With about 0 % success rates, the robot learns to reach the punishment side. As the learning progresses, the number of VTE increases and afterward decreases with the original model. In the case of the minimal model, we did not observe this VTE change.

**Going to the same direction** The robot learns to go to the same fixed side (right side or left side) and gets around 50 % success rates. The number of VTE remains high during the whole experiment.

**Behavioral transition** The robot transit among the three previous behaviors - going to the reward, the punishment, and the one side - and displays success rates between 30 % and 70%. This transition might have some relationship with chaotic itinerancy where the state of a system oscillates between different attractors. However, this behavioral transition cannot be seen in the minimal model. The number of VTE remains high during the whole experiment.

**Random** The robot acts seemingly randomly and the success rate is around 50 %. The number of VTE remains high during the whole experiment.

In order to investigate further the robustness of the evolved controllers against environmental change, we carried out the same experiments with different T-maze sizes. We varied the width and the length of the T-maze, as drawn in figure 1, and calculated the average and the variance of the success rates for every initial positions. With the original model, the robot does not change its performance in respect to the average and the variance of the success rates. The robot with the minimal model gets affected by a slight change in environmental size (figure 8). This result confirms that the presence of VTE can be an indicator of the robustness of the neural system.

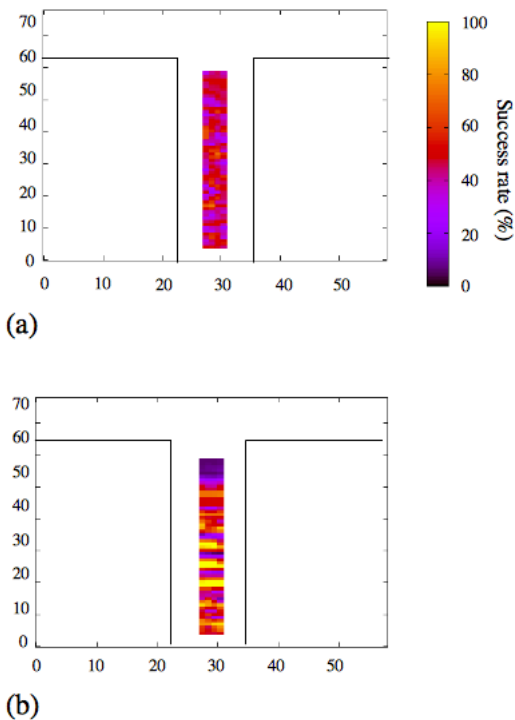


Figure 6: Average success rates for each starting positions. (a) In the case of the original model. (b) In the case of the minimal model.

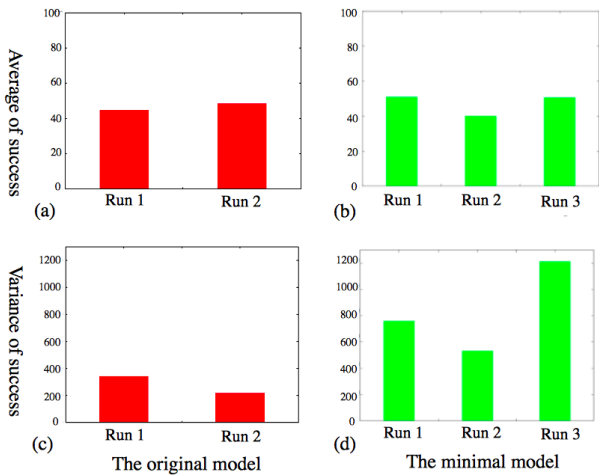


Figure 7: Average and variance of success rates per starting position. These graphs show the results of two (or three for the minimal model) GA runs of the original and the minimal model respectively. The red graphs present the results for the original model while the minimal model are in green. (a), (b) Average of success rates. (c), (d) Variance of success rates.



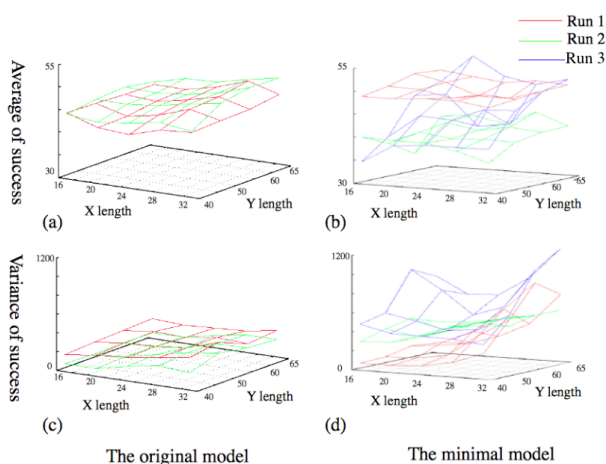


Figure 8: Average and variance of success rates per starting position, for different sizes of the T-maze. The size of the T-maze indicated by X and Y length corresponds to that of figure 1. Red and green lines corresponds to each two runs of GA. (a), (b) Average of success rates, for the original and the minimal model respectively. (c), (d) Variance of success rates.

## Conclusion

Our experiments aimed at uncovering the roles of VTEs through robotic experiments. Our work relies on a model developed by Bovet and Pfeifer where a neural network equipped with Hebbian learning commands a robot to complete a T-maze task using multiple sensory modalities (Bovet and Pfeifer (2005)). Unlike Bovet’s work, we composed the whole setup in a computer simulation, and conducted experiments under varied conditions, while optimizing the parameters of the model using a GA. This setup allows us to compute the number of VTEs during the learning of the route selection task.

We compared two models, one with full connectivity among modalities, and the other with minimal connectivity. Although both models exhibit the same performance, or 100% success, the former shows similar VTE curves to experiments with real rats, while the latter does not. This implies that VTE might not be related to performance in learning but would rather be caused by a redundant connectivity pattern.

We also noticed the original model, or the model with redundant connectivity, maintains its success rate to about 50% in most cases, regardless of perturbations to initial conditions or environmental size, which is accompanied by more VTEs. On the other hand, the model with minimal connectivity exhibits a lower robustness against perturbations. This model shows almost no VTEs. In conclusion, we offer the hypothesis that VTE might be linked to adaptivity to environmental changes.

In addition, we observed three seemingly stable behavioral patterns, and behavioral transition among those three patterns. This transition might have something to do with chaotic itinerancy (Ikegami (2007)). However, the dynamics of the neural network has not been studied and the cause of the VTEs has yet to be uncovered. Additional studies of this model, such as analyses based on chaos theory (Ogai and Ikegami (2008); Nakajima and Ikegami (2008)), or analyses from the field of differential topology (Thom (1972)), could shed some lights on the mechanisms of VTEs.

## Acknowledgements

We would like to thank three anonymous reviewers for their useful comments. We would also like to thank Simon Bovet for his help on reproducing the original experiments. Julien Hubert thanks the Monbukagakusho Scholarship from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

## References

- Bovet, S. and Pfeifer, R. (2005). Emergence of delayed reward learning from sensorimotor coordination. In *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, pages 841–846. Edmonton.
- Hebb, D. O. (1949). *The Organization of Behavior*. Wiley, New York.
- Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor : The University of Michigan Press.
- Hu, D. and Amsel, A. (1995). A simple test of the vicarious trial-and-error hypothesis of hippocampal function. *Proceedings of the National Academy of Sciences of the United States of America*, 92:5506–9.
- Ikegami, T. (2007). Simulationg active perception and mental imagery with embodied chaotic itinerancy. *Journal of Consciousness Studies*, 14:111–125.
- Johnson, A. and Redish, A. D. (2007). Neural ensembles in ca3 transiently encode paths forward of the animal at a decision point. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 27:12176–89.
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotocz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, volume 1, pages 59–65.
- Muenzinger, K. F. and Fletcher, F. (1934). Motivation in learning: I. electric shock for correct response in the visual discrimination habit. *J. Comp. Psychol.*, 17:266–277.
- Nakajima, K. and Ikegami, T. (2008). Dynamical systems interpretation of reversal of subjective temporal order due to arm crossing. *Adaptive Behavior*, 16:129–147.
- Ogai, Y. and Ikegami, T. (2008). Microslip as a simulated artificial mind. *Adaptive Behavior*, 16:129–147.

- Tarsitano, M. (2006). Route selection by a jumping spider (*Portia labiata*) during the locomotory phase of a detour. *Animal Behaviour*, 72:1437–1442.
- Thom, R. (1972). *Structural Stability and Morphogenesis*. W. A. Benjam.
- Tolman, E. C. (1939). Prediction of vicarious trial and error by means of the schematic sowbug. *Psychol. Rev.*, 46:318–336.



# Devil in the Details: Analysis of a Coevolutionary Model of Language Evolution via Relaxation of Selection

Luke McCrohon<sup>1</sup> and Olaf Witkowski<sup>2</sup>

University of Tokyo, Japan.

<sup>1</sup>luke.mccrohon@gmail.com

<sup>2</sup>olaf.witkowski@gmail.com

## Abstract

Computational modeling is an important tool in the study of language evolution. It is not only used to test hypotheses, but also as a source of data on difficult to observe evolutionary dynamics. This makes it particularly important to distinguish the emergent behaviors of evolutionary systems being studied, from the behaviors of specific models. In this paper we provide an in-depth analysis of one recent model of linguistic bio-cultural coevolution (Yamauchi and Hashimoto, 2010) and show that several of its reported behaviors are artifacts produced by the model's design and parameter settings. Specifically, we show that the model's population size setting and agent "geography" place strong limits on both cultural and biological diversity in the model. These limits interact with the model's learning mechanism and result in a number of semi-stable attractor states. We argue that it is the properties of these attractors that account for the long run behavior of the model, directly conflicting with the analysis given in the original paper. Our results are confirmed by experiments altering the model's population size parameter which result in a qualitative change in the observed model behavior.

## Introduction

The study of human evolution is complicated by the fact that in our species, phenotypes are shaped by the interaction of two separate evolutionary processes; biological evolution affecting our genes and cultural evolution affecting our learning environments. This *dual inheritance* (Richerson and Boyd, 2006) is perhaps most obvious in the study of human language, where despite the human ability to use language being biologically determined, the forms of the actual languages an individual acquires are determined by their cultural environment. The importance of this interaction diachronically, the so-called phenomenon of gene-culture coevolution, has in the last decade received growing recognition in the field of Evolutionary Linguistics (Deacon, 1997; Tomasello, 1999; Hurford and Kirby, 1999) and is also coming to be recognized in mainstream linguistics (Briscoe, 1998).

The most famous theoretical evolutionary gene-culture interaction is the Baldwin Effect (Baldwin, 1896; Simpson, 1953), a suggested process whereby initially learnt behav-

iors are gradually integrated into the genome. If the Baldwin Effect were in operation in language evolution it would work to increase the overall genetic contribution to the phenotype. Deacon (1997, 2003) however has suggested that language evolution is characterized by the opposite, a decrease in genetic contribution. He suggests that a relaxation of biological selection pressures, similar to that seen in domesticated animals, has given our lineage the evolutionary flexibility to evolve complex language. It has been argued that the cause of this relaxation of selection may have been through a cultural niche construction process (Odling-Smee et al., 2003; Yamauchi, 2004) in which cultural transmission was able to take over some of the burden of transmitting communicative behaviors between generations. This would have removed any selective pressure to keep these traits genetically hardwired, effectively allowing our ancestors to "self-domesticate" themselves via the culture they created.

Unfortunately, for anyone wishing to study the Baldwin effect, or other similar coevolutionary interactions, very little direct historical data exists. Biologically, the soft tissues on which our language ability depends seldom fossilize, and culturally, spoken words never do. This has led researchers to turn to a variety of indirect sources of data, such as comparative animal models, archaeological data, language acquisition studies and, recently, computational modeling. Not only are computational models being used to test hypotheses, but they are also being used directly as a source of data. This makes it particularly important that we understand the models that we are working with. Specifically, we need to be careful to determine whether any interesting dynamics we observe in our simulations are truly emergent properties of interactions of the target systems, or are just artifacts of our particular model designs.

In this paper we provide a detailed analysis of one recent exploratory computational model designed to investigate gene-cultural coevolution. The model was originally presented in Yamauchi and Hashimoto (2010) and we chose to investigate it due to its claimed cyclic repetition of stages in which biological selection was masked by cultural evolution, followed by stages in which biological selection was

vigorously reasserted. As such a cycle has not been at-tested in real world data, it was our intention to investigate its cause, and determine whether it was a product of an artificially high rate of simulated biological evolution when compared with the rate of cultural change. A possibility suggested by Chater et al. (2009), who argued that faster rates of culture change provide a moving target that biological evolution has a hard time adapting to.

However, on analysis, we show that the model's apparent cyclic behavior can be better described as a random walk between a linearly ordered set of attractor states. Furthermore, we show that the existence of these attractors is the result of arbitrarily chosen model parameter settings, and is not necessarily a consequence of properties of the target co-evolutionary system. We give a precise characterization of the set of attractor states and the transition probabilities between them and explain why the attractors exist and behave as they do. Finally, we show why the original set of parameters led to this behavior in the first place. We believe our analysis will prove useful to others interested in constructing gene-culture coevolutionary simulations and hopefully help prevent similar dynamics compromising future models.

This paper is structured as follows. The following section provides a description of the model we will analyze. This is followed by a summary of some key results from the original paper. We then describe our implementation of the model and where it can be obtained. This is followed by an in-depth analysis of the model's behavior based both on additional data from our implementation and a detailed study of the model's design. The final section then briefly discusses the implications of our analysis and highlights several points that need to be carefully considered during the construction of future coevolutionary models.

## Model of Language Evolution

In this section we describe the original model of Yamauchi and Hashimoto (2010).

In general terms, the model can be characterized as an agent-based language evolution simulation in which agent phenotypes are determined by a combination of their *biologically* inherited genomes and *culturally* transmitted knowledge. The model is designed to investigate the interactions between the biological and cultural aspects of agent evolution and is based on an earlier gene-culture coevolutionary model by Kirby and Hurford (1997). The main difference when compared with this earlier model is in the ways in which agents "learn", and specifically how their ability to learn is affected by their biological inheritance.

In Yamauchi and Hashimoto's model, each agent has a *chromosome* (a length 12 binary array), which represents its genetic predisposition towards learning each of 12 different linguistic alleles. For each allele the agent may be predisposed to learn either of its two possible forms; either zero or one. In addition to this chromosome, each agent also has a

*grammar* (a 12-value ternary array), representing its knowledge of its local language. For each allele the agent's grammar can either specify which form is used, zero or one, or may specify a lack of knowledge about the local language's instantiation of that allele, in which case its grammar will contain a *null* value. Additionally, agents also possess a certain supply of cognitive *learning resource*, initially set at 24 units.

The simulation proceeds via discrete generations, which each contain four phases. First, in the *Learning Phase* the current generation of agents are exposed to utterances from the previous generation and are given a chance to learn their grammars. Second, in the *Invention Phase* agents who still have null values in their grammar are given a chance to invent new values. Third, in the *Communication Phase* agents interact with their neighbors to determine their *fitness*. Finally, in the *Reproduction Phase* a new generation of agents is created via sexual recombination of agents from the current generation.

Prior to the beginning of the simulation, all agents in the current and previous generations have their chromosomes randomly initialized to zeros and ones. The grammars of both generations are also initialized to contain only nulls. This means that in the first generation of the simulation, agents will not receive any non-null inputs which they can learn from. Thus the initial chromosomes do not directly determine the initial culturally transmitted language of the agents.

Each generation contains 200 agents that are geographically arranged in a cycle. An agent's position in the cycle affects which agents it learns from in the Learning Phase and which agents it communicates with in the Communication Phase, but has no effect on the agent during the Reproduction Phase. That is, learning and fitness are determined locally on the cycle, but reproduction is determined globally. The motivation for this is discussed in Kirby and Hurford (1997).

In the Learning Phase agents are presented in turn with *utterances* taken randomly from members of the previous generation at a distance of at most two from the learner agent. Utterances are produced by looking at the source agents grammar and randomly selecting one of the 12 alleles. The utterance is then a pair of the allele index (1–12) and its value (zero, one or null). If the value of the utterance is null, it is instantly discarded and the learner agent moves on to the next utterance. Otherwise, the agent takes the utterance and compares its value with its own grammar at the specified index. If the value of the utterance is different from its own grammar, the agent attempts to update its grammar to match the utterance.

Updating the grammar requires the agent to have sufficient learning resource remaining to make the update. An update that sets the grammar to a value equal to the value of the agents chromosome at the same index costs the agent 1

unit of learning resource, an update to a value different from the agent's chromosome costs the agent 4 units of learning resource. If sufficient learning resource is available, the cost is subtracted from the agent's supply of learning resource and the update is made. If the agent has insufficient learning resource, the agent's learning resource is set to 0, and no update is made. For each agent, the learning phase continues until either it has been exposed to a total of 200 utterances or until its learning resource is depleted.

Once the Learning Phase has been completed for all agents, the Invention Phase begins. All agents that have learning resource left are given a chance to use that resource to fill in any null values left in their grammars. For each left over unit of learning resource an agent has a 0.01 chance of filling a single null with a randomly selected value of zero or one. Once either the learning resource is exhausted or no nulls remain in the grammar, invention stops, and the grammar of the agent is fixed for the remainder of the simulation.

Next the simulation enters the Communication Phase. This phase serves to establish the fitness of each agent, which will be used to determine their likelihood of contributing to the next generation of agents in the Reproduction Phase. Initially each agent has its fitness initialized to a base value of 1. Beyond this, fitness is determined by the ability of the agent to successfully communicate with its neighbors. Each agent has 6 chances to communicate with each of its immediate neighbors on the cycle in the same generation.

To determine if the agents are successful in communicating, one of the 12 grammatical alleles is randomly selected and the grammars of both agents are compared at that allele. If either agent has a null value for that allele, or if the two agents disagree on its value, then communication is declared a failure and the agents' fitness is left unchanged. However, if both agents' grammars agree, communication is declared a success and both agents receive plus 1 to their fitness score. When this process is complete all agents will have been assigned a fitness score of between 1 and 13.

Finally, the simulation enters the Reproduction Phase in which the next generation of agents is produced. To create each new agent, two parent agents are selected from the current generation via roulette wheel selection. The location of agents on the cycle is ignored for the purposes of their selection. The chromosomes of the two parent agents are then combined via obligatory single point crossover. Then with a probability of 0.00025 each allele is mutated. This chromosome is then used to create a new agent which is initialized with an empty grammar (all null).

Once reproduction is completed, the current generation is replaced by the new generation and the above four phases are repeated. For a more complete description of the model please refer to the original paper or examine the freely available source code of our implementation.

Before proceeding to the results that this model produces, we would like to note that, as is fairly common in this type of

modeling, the majority of parameter values are set to largely arbitrary values. That is to say there is no external linguistic factor motivating setting the population size to 200 or in setting an agent's initial learning resource at 24 units, they just happen to be values that seem to work and produce the intended dynamics.

## Simulation Results

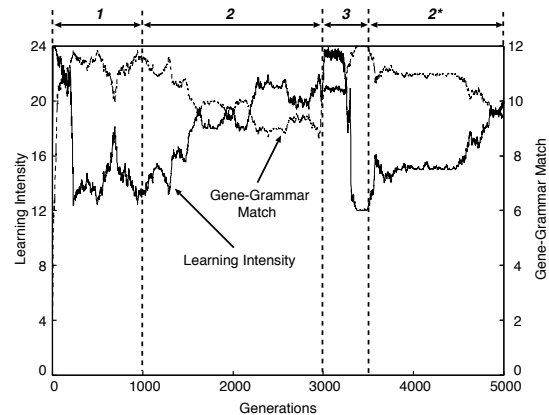


Figure 1: Gene-Grammar Match in the Original Model<sup>1</sup>

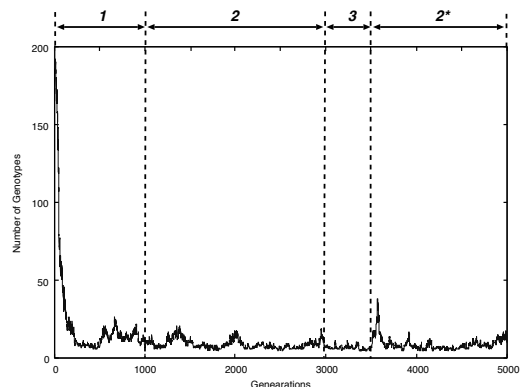


Figure 2: Number of Genotypes in the Original Model<sup>1</sup>

Two key charts from Yamauchi and Hashimoto (2010) are reproduced as figures 1 and 2. The first shows the progression of the Gene-Grammar match (average hamming distance between mature agents' grammars and chromosomes) and the learning intensity (average amount of learning resource consumed by agents in the Learning Phase). The second shows the number of different genotypes present in the population over time. The original paper divides the discussion of these results into three stages as marked by the dashed vertical lines in the figures.

<sup>1</sup>Reproduced with permission of the original authors.

### Stage 1 - Baldwin Effect

The first stage spans the first few hundred generations and covers the period in which agents go from initially having no culturally transmitted language (and a very low fitness) to having a highly uniform language shared between all agents (and consequently maximal fitness). The language that results from this phase is shown to match the grammar to an above chance level, and by the end of the phase, the genetic diversity has decreased significantly. This is claimed to be the result of an assimilatory process akin to the Baldwin effect, operating to allow agents to save more learning resource for the invention of new tokens to replace any null elements in their grammars.

### Stage 2 - Functional Redundancy

The second stage takes place over several thousand generations throughout a period in which biological selection is masked by a culture. Following stage 1, the culturally transmitted language closely matches the agents' innate biases and provides them a stable uniform stimulus to learn from. This simplifies learning and allows all agents to successfully learn a single common language. As a result there are no problems in communication, and all agents are assigned the maximum fitness score. Biological selection has been effectively masked by culture. Throughout this stage the culturally determined language transmitted between generations remains largely unchanged. The masking of biological selection leads to a relaxation of selection on the agents' biases (the biases have been made functionally redundant), and they are free to degrade to values that no longer match the culturally transmitted language. The original paper claims that as a result of this "the degree of correlation between the gene-pool and the environment gradually, yet firmly declines" throughout this stage.

### Stage 3 - Unmasking of Natural Selection

This stage begins when the gene-grammar match has deteriorated to a point at which biological natural selection is no longer masked and biological selection again begins to take effect. It is claimed that this results in agents in a local area converging on different I-languages which decreases their fitness and cause problems for agents in the subsequent generation to learn the language. Due to this, a biological assimilatory process begins to take affect, similar to that seen in stage 1, which quickly returns the population to a point with high gene-grammar matches as was present at the beginning of stage 2. Having returned conditions to how they were at the onset of stage 2 it is claimed that stages 2 and 3 then repeat cyclically every few thousand generations.

### Our Implementation

Unfortunately the source code of the original model is not publicly available and so to investigate it further we reimplemented it ourselves following the details given in the orig-

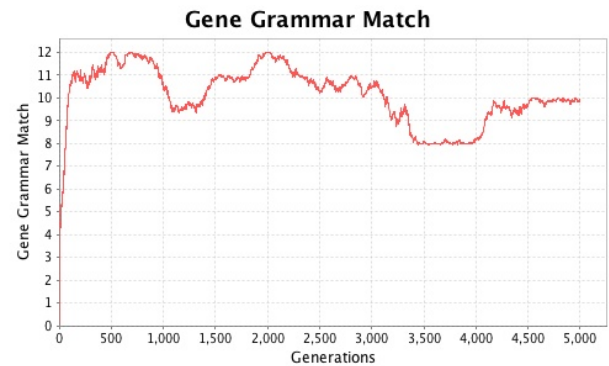


Figure 3: Gene-Grammar Match in our implementation (c.f. fig.1) [Seed=1303050913721, Runs=1, Generations=5000]

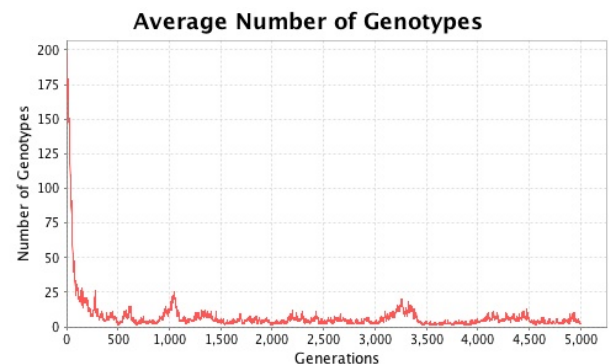


Figure 4: Number of Genotypes in our implementation (c.f. fig.2) [Seed=1303050913721, Runs=1, Generations=5000]

inal paper. The specifications were sufficiently precise that our implementation produces results which closely mirror those reported in the original paper. For comparison we present figures 3 and 4 which show the same range of behaviors as those of the original model depicted in figures 1 and 2.

In presenting our own results we generally concentrate on only the gene-grammar match, ignoring the learning intensity. This is because except for the initial generations, where there is a significant number of nulls, the learning intensity is essentially a scaled inverse of the gene-grammar match. We also tend to ignore the number of nulls, fitness, number of genotypes, et cetera, as for the most part they tend to produce relatively stable values throughout the simulations.

Our implementation of the model is available for download at:

<http://code.google.com/p/suzume/>

To run the model as described in this paper select "YamauchiHashimoto2010" as the Agent Type, "CyclicBag-Model" as the Population Model and "RouletteWheelSelection" as the Selection Model. The version of our code used

to conduct the experiments presented in this paper was Rev ac9d4c742fe2.

Unless otherwise specified, all simulation results presented in this paper were conducted with the default configuration parameters settings. Each result set is presented together with the random generator seed, run count and generation count used to produce it. This information should suffice to reproduce the data underlying all figures presented in this paper.

## Analysis

### Genetic Diversity

One of the features that led the authors of the original paper to conclude that stage 1 was the result of a Baldwin Effect style assimilatory process was the overall decrease in genetic diversity during this stage. But as figure 5 shows, even when selection has been set to ignore agent fitness values (resulting in neutral biological selection), the same reduction in genetic diversity is observed. This reduction is caused by genetic drift in the relatively small population fixing random alleles. This process brings the overall number of genotypes down to approximately 5-10. The number of genotypes never reaches 1 because new variations are constantly being introduced by mutation. 5–10 is the level at which new variants are being introduced by mutation at the same rate as which they are being removed by drift.

This process of drift removing variation and mutation adding it, continues throughout the simulation and results in a rather constant level of genetic variation despite the absence of biological selection in stages 2 and 3. The level at which the number of genotypes stabilizes can be altered by changing the mutation rate of the agents or by changing the population size (smaller populations are more easily affected by drift fixing values). Looking at just the number of genotypes actually makes the degree of variation in the population seem greater than it actually is. While there may be 5-10 variants at any given time, it should be noted that these are normally very closely related to each other and are usually only represented by a small number of individuals.

What the decrease in variation from random drift implies is that the initial reduction in genetic variation observed during the first stage should not be seen as evidence of an adaptive genetic process such as the Baldwin Effect.

### Masked Genetic Selection

As was reported by the authors in the original paper, genetic selection is effectively masked in stages 2 and 3 of the simulation (approximately 1000 generations onwards). And so we should expect genotypes to take a random walk through the space of possibilities for as long as selection remains masked. If we look at figure 6, which shows 10 separate runs over 10000 generations, we see this does in fact occur to a certain degree, but the gene-grammar matches never drop below 8 for any significant period of time. This would

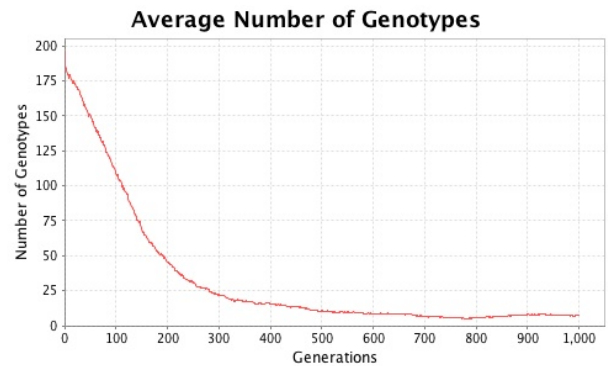


Figure 5: Genotype Variation over initial 1000 generations under neutral biological selection (c.f. fig.2) [Seed=1302966692486, Runs=50, Generations=5000]

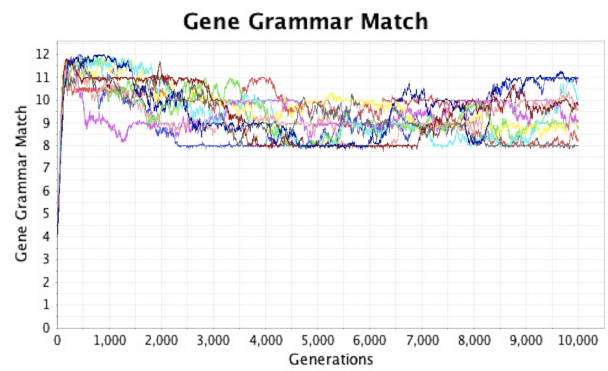


Figure 6: Gene-Grammar Match for 10 independent runs over 10000 generations [Seed=1303127096921, Runs=10, Generations=10000]

suggest that this is the limit to which cultural shielding can operate. That this is in fact the case, can be argued directly from the model design.

There are two ways in which an agent's learning resource can be depleted. First, an agent may be exposed to conflicting inputs which causes it to repeatedly switch the value of a grammatical allele, quickly depleting the resource. But as our experiments and the results of the original paper show, this is unlikely to occur once a common language has been established, as that language is very stable. Alternatively, this may occur if an agent is presented with consistent input, but when that input is so divergent from its innate biases that the learning cost is greater than the agents initial learning resource supply. The value at which this occurs can be calculated based on the simulation parameters.

Agents begin with 24 units of learning resource and need to fill the 12 alleles in their grammar. The cost of filling an allele that matches their chromosome is 1, and the cost of filling a non-matching allele is 4. Therefore the maximum

number of non-matching alleles the agent can learn while successfully filling its chromosome is four, any more and it won't have sufficient resources left to fill the remaining alleles ( $4 \text{ non-matching} \times 4 + 8 \text{ matching} \times 1 = 24$ ). Experiments changing learning costs and the agents supply of learning resource alter the shielding level as expected.

This means any agent dropping below a gene-grammar match of 8 will not be able to fill its grammar, and so would have its fitness penalized and would be selected against by biological selection. This is different to the reasoning presented in the original paper where it was suggested that at this point the agents would be selected against due to increasing variation in their learning input. This will occur in subsequent generations with agents being subjected to null inputs, but is not the original cause of the unmasking of biological selection.

### Coevolutionary Attractors

A close inspection of figure 6 shows that there are certain values at which the gene-grammar match occurs more frequently. Specifically those values centered around integer values between 8 and 12. This can be seen clearly in the probability density plot shown in figure 7.

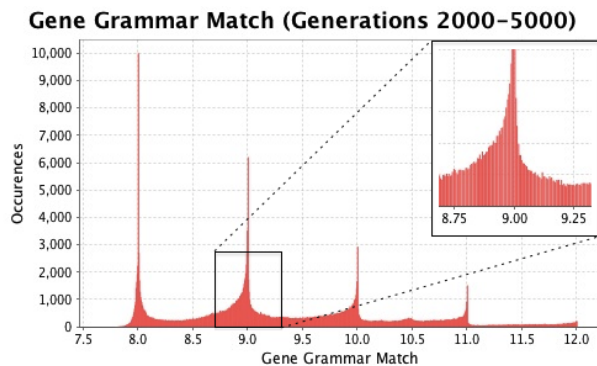


Figure 7: Gene-Grammar Match Density [Seed=1303046232707, Runs=100, Generations=5000]

The reason the gene-grammar match occurs most frequently around these values is due to the previously mentioned facts that the language is uniform across agents, and that genetic variation is highly limited. If there is only one language, and if the vast majority of agents share the same genes, then the average gene-grammar match will fall close to an integer value. It is only when significant portions of the population possesses different genes that the population will move away from these points. In cases where this does happen, genetic drift will usually sweep the population back to its original integer value point. In rare cases however, if the population moves sufficiently far away from its previously stable genetic state, drift may cause the population to be swept to a new uniform genetic state (and hence a differ-

ent integer gene-grammar match value).

Of course there may be several different chromosome-grammar matches that result in agents exhibiting the same gene-grammar match value. However, as nothing in the agent's learning algorithm changes their probability of learning individual grammatical alleles due to a particular set of genetic biases (only the number of matches ultimately influences learning), these different model states will behave identically. Because of this it is safe to view the integer value gene-grammar matches as attractor states in the simulation, despite them potentially representing a number of different underlying gene-culture states.

	to 12	to 11	to 10	to 9	to 8
from 12	.55	.05	—	—	—
from 11	.01	.52	.07	.01	—
from 10	—	.02	.42	.08	—
from 9	—	—	.02	.61	.03
from 8	—	—	—	.04	.78

We calculated the likelihood of the simulation jumping between each of these attractor states ( $\pm 0.2$  units) over a period of 200 generations. The transition probability matrix is presented in the table above and in the transition diagram in figure 8. We tested these results against transitions between equally sized intervals positioned directly between the attractor states and obtained probabilities of the simulation staying in those intervals approximately 5 times lower than in the case of the attractors. This indicates that the attractors are significantly more stable.

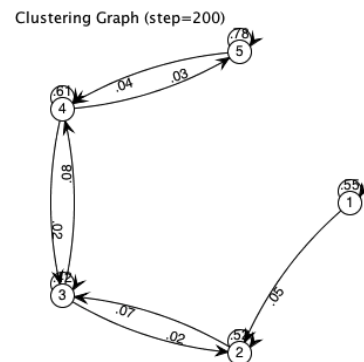


Figure 8: State Transition Diagram [Seed=1303037425613, Runs=50, Generations=20000]

### Shape of the Attractors

As can be seen in figure 7 the attractors are not symmetrical. For all attractors except the lowest one (at a gene-grammar match of 8) there are significantly more values in the region directly below them, than in the region above. This is representative of the fact that deviations from the fixed point are more likely to be in a downward direction. This



happens because these deviations are the result of biological changes, and as the four attractors in question (at gene-grammar matches of 9, 10, 11 and 12) all represent genotypes in which more than half of the grammar alleles match the agent's biases. Thus the majority of random changes to agent's genotypes will result in a decrease in the gene-grammar match.

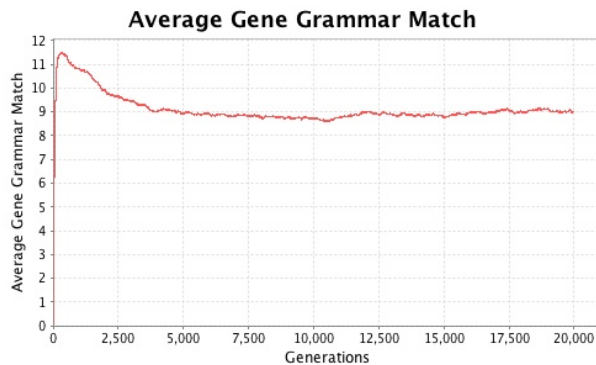


Figure 9: Average Gene-Grammar matches [Seed= 1303037425613, Runs=50, Generations=20000]

This results in a greater probability of downward transitions between attractors, and in the long run attractors results in the lower attractors having a greater chance of being occupied. This long run effect is visible in figure 9. For approximately the first 5000 generations the model has yet to settle down following the high gene-grammar matches attained in the first few hundred generations.

### Attractors and Population Size

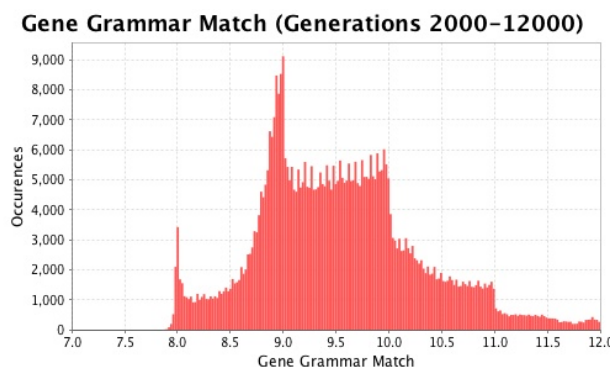


Figure 10: Gene-Grammar matches for a large population of 400 agents [Seed= 1303033229645, Runs=50, Generations=12000]

As the culturally determined language is largely stable throughout the simulation, the primary requirement for falling into an attractor is that all agents show very little

genetic variation (if there is significant variation the average gene-grammar match is unlikely to approach an integer value). As was discussed earlier the main reason the simulation generally ends up in states with little genetic variation is due to genetic drift sweeping away what variation that does exist. Unsurprisingly this is more of an issue with a smaller population, and our experiments show that the lower the population size, the less genetic variation exists, and the more defined the fixed points become.

Conversely, increasing the population size makes the fixed points less distinct as can be seen in figure 10 which shows gene-grammar occurrence frequencies in a population of 400 agents. What is surprising is when we increase population size further still (as shown in figure 11 with a population of 1000 agents), the fixed points disappear entirely. With the fixed points removed, the simulation loses the downward ratchet effect caused by the shape of the fixed points and demonstrates a qualitatively different behavior than was seen with smaller population sizes. We think this is likely the behavior that was originally intended by Yamauchi and Hashimoto (2010). Unfortunately an analysis of the dynamics of the model with this larger population size is outside the scope of this paper.

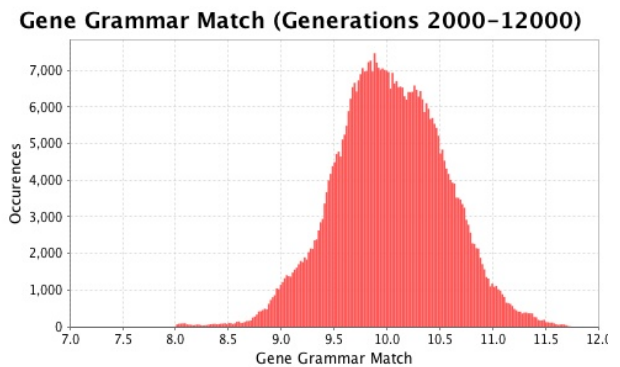


Figure 11: Gene-Grammar matches for a very large population of 1000 agents [Seed= 1302981824159, Runs=50, Generations=12000]

### Conclusions and Discussion

While the model presented in Yamauchi and Hashimoto (2010) does reproduce many of the interactions it sets out to capture (Cultural Shielding, Niche Construction etc.), the behavior of the model within the limits set by these interactions has been shown to be the result of the model's design, and not of any underlying emergent properties of its target system (e.g. gene-culture coevolution). We have not provided evidence directly contradicting any of the conclusions reached based on the original model, but have shown that additional work is necessary to understand the dynamics of the system that was investigated. We think that the analysis



presented in this paper will be useful to others seeking to design coevolutionary simulations, particularly as inspection of charts related to the coevolutionary simulation of Kirby and Hurford (1997) also suggests evidence of a similar sort of fixed points to those identified here.

**Population Size is Significant** Population size usually only has a quantitative effect on simulation behavior, but as was demonstrated in our analysis, under certain circumstances, it can have a significant qualitative effect. Given that increasing population size can be computationally expensive, we think that it might be sensible to investigate alternative population structures that may be able to imitate the behaviors of larger populations. At a minimum we think our analysis demonstrates the necessity of testing coevolutionary simulations on a wide range of population sizes to see if there exist any qualitative effects.

**Cultural Diversity is Important in Coevolution** Many of the dynamics we identified in this model were the result of it only supporting limited cultural diversity. The fixed points we identified would not have been present if the model had been able to support multiple culturally determined languages concurrently. Additionally, when the model strayed away from the zone in which biological evolution was shielded by culture, and biological selection reasserted itself, had there been more than a single language present the dynamics would have likely proved more interesting. The lack of cultural diversity/change in the present model can be traced directly to its learning mechanism, when learning resources are sufficient as they are throughout the majority of the simulation, there is a near zero probability of cultural change occurring. Future simulations should be designed to allow at least a certain level of diversity, not just in their representation of biology, but also of their culture.

**Biological Selection Geography** Running roulette wheel selection on a relatively small population of 200 agents in which there is no concept of distance resulted in low levels of genetic diversity throughout the simulation. Genetic diversity was shown to be raised by increasing the population size which effectively removed the fixed points seen in the simulation, but at the cost of a heavily increased computational burden. We suggest that future simulations could take advantage of alternative selection methods such as trimming some fraction of poorer performing agents from the reproductive population as in Kirby and Hurford (1997). Or alternatively by adding some concept of geography into the replacement algorithm.

**Random Walks in a Binary Space** During Stage 2 of the original paper's analysis, the authors claim that the relaxation of selection leads to a gradual yet firm degradation in the gene-grammar match of the agents. But as we have

shown, this is in fact simply a random walk which also has the possibility of increasing the match. If instead of using a binary space of language possibilities the authors had of used a higher dimensional space, the random nature of the walk would have been far less likely to increase the gene-grammar match, and would have better demonstrated the degradation dynamic that was intended.

## Acknowledgements

We would like to thank Hajime Yamauchi for permission to reproduce the charts from (Yamauchi and Hashimoto, 2010). This paper was partially supported by Grant-in-Aid for Specially Promoted Research (MEXT, Japan).

## References

- Baldwin, J. M. (1896). A new factor in evolution. *The American Naturalist*, 30(354):441–451.
- Briscoe, T. (1998). Grammatical acquisition: Coevolution of language and the language acquisition device. In *In Proceedings of the Diachronic Generative Syntax*. Oxford University Press.
- Chater, N., Reali, F., and Christiansen, M. (2009). Restrictions on biological adaptation in language evolution. *PNAS*, 106(4):1015–1020.
- Deacon, T. W. (1997). *The Symbolic Species: The Co-evolution of Language and the Brain*. W.W. Norton.
- Deacon, T. W. (2003). Multilevel selection in a complex adaptive system: The problem of language origins. [References]. In A., Division, Department, and Anonymous, editors, *Evolution and Learning: The Baldwin Effect Reconsidered*. *Life and mind*, pages 81–106. The MIT Press.
- Hurford, J. and Kirby, S. (1999). Co-evolution of language size and the critical period. In Birdsong, D., editor, *Second Language Acquisition and the Critical Period Hypothesis*, pages 39–63. Lawrence Erlbaum.
- Kirby, S. and Hurford, J. (1997). Learning, culture and evolution in the origin of linguistic constraints. In Husband, P. and Harvey, I., editors, *ECAL97*, pages 493–502. MIT Press.
- Odling-Smee, F., Laland, K., and Feldman, M. (2003). *Niche construction: the neglected process in evolution*. Monographs in population biology. Princeton University Press.
- Richerson, P. J. and Boyd, R. (2006). *Not by Genes Alone: How Culture Transformed Human Evolution*. University Of Chicago Press.
- Simpson, G. (1953). The baldwin effect. *Evolution*, 7:110–117.
- Tomasello, M. (1999). *The cultural origins of human cognition*. Harvard University Press.
- Yamauchi, H. (2004). *Baldwinian Accounts of Language Evolution*. PhD thesis, Theoretical and Applied Linguistics, University of Edinburgh, Scotland.
- Yamauchi, H. and Hashimoto, T. (2010). Relaxation of selection, niche construction, and the baldwin effect in language evolution. *Artificial Life*, 16(4):271–287.

# Evolution of Self-Maintaining Cellular Information Processing Networks

Barry McMullin<sup>1</sup> and James Decraene<sup>2</sup>

<sup>1</sup>The Rince Institute, Dublin City University, Ireland

<sup>2</sup>School of Computer Engineering, Nanyang Technological University, Singapore  
barry.mcmullin@dcu.ie

## Background

This contribution is an extended abstract of (Decraene and McMullin, 2011). What we here term Cellular Information Processing Networks (CIPNs) are biochemical systems of interacting molecules occurring in living cells. CIPNs are responsible for coordinating cellular activities in response to internal and external stimuli (e.g., chemotaxis signalling pathways). CIPNs can be regarded as special purpose computers (Bray, 1995). A single enzyme molecule effectively carries out pattern matching to identify and bind target substrate(s), and then executes a discrete computational operation in transforming these into the product molecule(s). The concept of *collective autocatalysis*, formulated by Farmer et al. (1986), denotes a collection of molecular species where each of them is the product of at least one reaction catalysed by at least one other species of the set. Fontana and Buss (1994) developed this into a more general formal concept of (collective) *self-maintenance*, and it has more recently been elaborated and refined in the *Chemical Organization Theory* of Dittrich and Speroni (2007). Self-maintenance ensures that reaction networks can reconstitute themselves when subjected to perturbations and during cellular divisions. It may thus mediate between the conflicting objectives of robustness and evolvability in reaction networks.

In contrast to modern living cells, the cellular model considered here does *not* incorporate a distinct genetic translation system. It is motivated by the (presumed) evolution of information processing in (proto-)cells *prior* to the emergence of the genetic architecture.

## The Artificial Chemistry (MCS.bl)

We employ an agent/string-based Artificial Chemistry called the Molecular Classifier System (MCS.bl<sup>1</sup>) which is based on Holland's broadcast language (Holland, 1992, pp. 143-152). The basic elements (the abstract "molecules") are formally strings on a specified symbol alphabet ("atomic" species). Chemical reactions are stochastic (molecular "mutation" may alter the generated product strings), reflexive

(no distinction made between operands and operators) and catalytic. Any single molecule may contain several condition/action rules which define its binding and enzymatic properties. In general the broadcast language allows arbitrary string transformations (computations) to be expressed; however, for the specific experiments described here, individual autocatalysis (self-catalysed replication) is explicitly disabled. Populations of molecules are encapsulated in containers to form "cells"<sup>2</sup>. Each cell functions as a separate well-stirred reactor. The number of molecules in a cell may increase until the cell matches a specified division criterion; a cell then divides with stochastic assortment of molecules into two daughter cells. Where particular molecular species are present in small numbers in a parent cell they may be absent completely in one of the two daughter cells, giving rise to distinct, cell-level, mutation events. The total number of cells is fixed: each division triggers the removal of another cell selected at random. The system is implemented on a small parallel computer cluster, with one CPU per cell. The real-time required for individual molecular interactions may vary with the specific detailed structures of the molecules involved. Cell reproduction rate is dependent on the real-time rate of catalytic reactions occurring in the cell, and on the specific criteria in effect for cell division. Distinct, interacting, selectional dynamics arise at both molecular and cellular levels.

## Experiment: Molecular Amplification

In the first experiment cells are evolved to carry out amplification of a given molecular species. This is motivated by conceptually similar *in vivo* investigations reported in the literature (Fong et al., 2005). The cell division criterion is configured so that cells divide when a target molecular species ( $s_T$ ) reaches a threshold number of instances. The cellular reproduction rate (fitness) therefore depends on the ability of the cell to promote the growth of  $s_T$  while still preserving overall collective self-maintenance of all required molecular species. The system is initialised ("seeded") with

<sup>1</sup>MCS.bl source code and documentation is available at: <http://esignet.net/dokumente/upload/WP13>

<sup>2</sup>For brevity, we say simply "cell" here; but this should be read as "proto-cell" throughout.

a hand-designed, viable cellular species (self-maintaining at the molecular level, including the target molecular species, so that cellular division is possible).

Similar phenomena are encountered in multiple runs. One typical run is described and analysed in more detail. In the course of this run, 1235 different and unique cellular species were generated in total, but of these, just three successively came to dominate the cellular population, through three identifiable displacement events. Careful analysis of both the molecular dynamics within the dominant cellular species and the cell-level population dynamics allowed determination that the first observed displacement in this run was selectional, with a clear increase in fitness (molecular amplification function); but the subsequent two displacements represented drift among essentially equal-fitness cellular species. That said, more fine-grained examination of the displacement events also shows that they are correlated with significant transient increases in cellular species diversity. These displacements are thus significantly more intricate than straight selection or drift between two “pure” cell lines. In effect, a single molecular mutation event can give rise to a complex cascade of cell-level mutations.

### Experiment: Crosstalk

Crosstalk phenomena arise very naturally in real biochemical information processing networks due to the fact that molecules from different signalling pathways may share the same physical reaction space (the cell). Depending on the relative specificities of the reactions there is then an automatic potential for any given molecular species to contribute to signal levels in multiple pathways. Here we describe an experiment investigating the evolutionary dynamics arising when distinct cells, with potential crosstalking pathways, are forcibly merged, but subsequent cell division is constrained to maintain selected molecular components from *both* pre-existing reaction networks (so cellular species in which one network simply displaces the other cannot continue to reproduce). This work is naturally related to the symbiogenesis theory which was originally postulated by Mereschkowsky (1910), and already explored computationally by Barricelli, on the first stored program digital computers, in the 1950's (Barricelli, 1957).

Over a number of runs, various common features are observed under these experimental conditions. A very rich variety of cellular species emerges, and in general there is significantly more cellular species diversity than in the previously described experiment: it is rare in this case for a single cellular species to exceed more than half of the population. Nonetheless, distinct displacement events can still be observed; and it is possible to analyse the molecular behaviour of a selection of mutant cellular species in detail. It is typical to observe the emergence of cellular species containing a “meta-reaction network”, still including all the seed molecular species, but also additional molecular species, exploiting

crosstalk and bridging between the seed species, and participating in the collective self-maintenance. In this sense, this experiment demonstrates a some (limited) evolutionary growth in the complexity of the self-maintaining reaction networks — both in terms of number of species and number of reactions composing the network. It is also observed that the gestation time of the dominant cellular species successively decreases

### Conclusion

We have presented a preliminary investigation of the role of collective self-maintenance in the evolution of (proto-)cellular information processing reaction networks. To assist this research, we built a novel agent-based multi-level selectional Artificial Chemistry. This was applied to the evolution of a single and multiple/crosstalking self-maintaining reaction networks. In these experiments, cellular species were successfully evolved to achieve the pre-specified information processing functions more effectively and exhibited a relatively higher level of complexity (by at least some reasonable measures). This proof of concept should contribute, to some extent, to understanding of the much more general problem of *open-ended* evolutionary growth of complexity using Artificial Chemistries.

### Acknowledgements

This work was supported by the ESIGNET Project (Evolving Cell Signalling Networks *in silico*), funded under the EU FP6 NEST initiative, contract no. 12789.

### References

- Barricelli, N. (1957). Symbiogenetic Evolution Processes Realized by Artificial Methods. *Methodos*, 9(35-36):143–182.
- Bray, D. (1995). Protein Molecules as Computational Elements in Living Cells. *Nature*, 376(6538):307–312.
- Decraene, J. and McMullin, B. (2011). The evolution of complexity in self-maintaining cellular information processing networks. *Advances in Complex Systems*, 14(1):55–75. Available from: <http://doras.dcu.ie/16292/>.
- Dittrich, P. and Speroni, P. (2007). Chemical Organisation Theory. *Bulletin of Mathematical Biology*, 69(4):1199–1231.
- Farmer, J., Kauffman, S., and Packard, N. (1986). Autocatalytic Replication of Polymers. *Physica D*, 22(1-3):50–67.
- Fong, S., Burgard, A., Herring, C., Knight, E., Blattner, F., Maranas, C., and Palsson, B. (2005). In Silico Design and Adaptive Evolution of *Escherichia coli* for Production of Lactic Acid. *Biotechnology and bioengineering*, 91(5):643–648.
- Fontana, W. and Buss, L. (1994). “The Arrival of the Fittest”: Toward a Theory of Biological Organization. *Bulletin of Mathematical Biology*, 56(1):1–64.
- Holland, J. (1992). *Adaptation in Natural and Artificial Systems*. MIT Press Cambridge, MA, USA, 2nd edition. See also Holland(1975).
- Mereschkowsky, C. (1910). Theorie der zwei Plasmaarten als Grundlage der Symbiogenesis, einer neuen Lehre von der Entstehung der Organismen. *Biol. Centralbl*, 30:278–288.

# Digital Sex: Causes and Consequences

Dusan Misevic<sup>1</sup>, Charles Ofria<sup>2,3</sup> and Richard E. Lenski<sup>2,4</sup>

<sup>1</sup>INSERM U1001, Centre de Recherches Interdisciplinaires (CRI), Université Paris Descartes, Paris, France

<sup>2</sup>Program in Ecology, Evolutionary Biology, and Behavior, Michigan State University, East Lansing, MI, USA

<sup>3</sup>Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, USA

<sup>4</sup>Department of Microbiology and Molecular Genetics, Michigan State University, East Lansing, MI USA

dulse@alife.org

## Abstract

Many theories have sought to explain the evolution of sex, but the question remains unanswered owing to the scarcity of compelling empirical tests. Here we summarize the results of two of our published studies investigating the evolution of sex using digital organisms. We used these evolving programs to test the hypothesis that sexual reproduction is advantageous in changing environments. We found that sex evolved to be the dominant mode of reproduction only when the environment was changing rapidly and substantially. Additionally, we measured the effects of sexual reproduction on genetic architecture, specifically modularity and epistasis. We found that sex profoundly influences genome organization, increasing modularity and decreasing the effects of interactions between mutations. Our studies have contributed to understanding both the causes and consequences of sexual reproduction, while also demonstrating the efficacy and power of *in silico* approaches to these issues.

## Introduction

Why sex? The paradox of sexual reproduction – a process that is costly and complicated, yet widespread in nature – has fascinated biologists for well over a century, and has in turn generated a wide range of hypothesis and experimental tests [1-3]. One of the simplest and perhaps most intuitive explanations is that sex accelerates the rate of adaptation to novel or changing environments by increasing genotypic and phenotypic variation [4]. Here we summarize a previously published study testing this theory *in silico* [5] as well as another study examining the effects of recombination on genetic architecture [6].

## Methods

All experiments were conducted using Avida software (freely available at <http://avida.devosoft.org/>), previously used in many studies of evolutionary trajectories and outcomes [7-8]. Digital organisms in Avida are short self-replicating computer programs that mutate, evolve, and reproduce either asexually or sexually, depending on which divide instruction they execute. Genomes were built from the default instruction set with 27 instructions including 2 divide instructions, divide-sex and divide-asex, only one of which can be expressed by any individual. In these studies, point, insertion, and deletion

mutations occurred at rates of 0.002, 0.0005, and 0.0005 per instruction copied, respectively, with the same mutation rates applied to the divide instructions as all others. When a population was at its carrying capacity (here 3600 organisms), each new offspring replaced a randomly chosen organism. All experiments ran for 100,000 updates (the Avida time unit), and a generation typically required 5–10 updates, with the precise number depending on the organisms' genomic and phenotypic complexity.

**Digital metabolism.** An organism's genome may contain instructions that encode the ability to metabolize one or more substrates present in the environment. Metabolism of a substrate either accelerates or decelerates an organism's replication rate by a factor of  $2^m$ , where  $m$  is the substrate's metabolic value and is positive or negative, signifying a nutrient or a poison, respectively. Fitness is calculated as the organism's total energy (energy obtained via metabolism in addition to basal energy provided equally to all organisms) divided by the time used to produce an offspring.

**Environmental conditions.** For the study of the effects of sexual versus asexual reproduction on genetic architecture, we evolved populations in a constant environment with 9 substrates that were always available in unlimited amounts. When testing the possible benefit of sex in changing environments, we used the same constant environment for the first 1000 updates of each experimental run, after which additional and changing substrates were introduced.

**Recombination mechanism.** Recombination is initiated by pairing up the genomes of two progeny that were produced sexually (i.e., divide-sex was expressed) and consecutively. The pair then exchanges a single continuous genomic region. The recombining region is chosen at random, but is matched between the organisms based on its relative position in the genomes. After recombination, both offspring are placed at random locations in the population, in the same manner as asexually produced organisms. The Avida mechanism of recombination (see [9] for a more detailed explanation) differs from others presented elsewhere in the Artificial Life literature. For example, in Tierra, sex involved recombination between living and deceased organisms [10], while in another system recombination somewhat resembled plasmid transfer [11]. Moreover, those studies were not driven by hypothesis

testing, but rather were descriptive and phenomenological in scope, making any comparisons difficult.

## Results

### Effects of changing environment on reproductive mode.

The trajectories of the relative abundance of sexual and asexual organisms were highly variable during our experiments. Overall, asexual reproduction prevailed, except at the highest rates of environmental change, when sexual reproduction tended to be more common. This result was obtained both when comparing the final mode of reproduction and when measuring the time that populations spent as predominantly sexual or asexual over the course of their evolution.

**Origin versus maintenance of sex.** Given the costs of sexual reproduction, it may be easier to maintain sex than to evolve it *de novo* [12]. We found that over the entire duration of the experiment, the populations started with sexual ancestors were predominantly sexual 38% more often than those with asexual ancestors. However, when considering only the latter half of the experiment, this difference was reduced to 25%, indicating the time necessary to make the switch between the modes of reproduction also played an important role. Overall, sex overcame the barriers that hindered its establishment in previously asexual populations only about half the time even under the most favorable treatments.

**Mode of reproduction and modularity.** We conducted extensive mutational analysis of organisms randomly sampled from populations that evolved in a constant environment with either obligatory sexual or obligatory asexual reproduction. We found that sexual organisms evolved to have both higher physical modularity (shorter distance between the genomic sites encoding a computational trait) and higher functional modularity (less overlap between the sites that encode two or more traits) than asexual organisms.

**Mutational sensitivity and epistasis.** Sexual populations also evolved to be significantly more robust to individual mutations than the asexual populations. Under both modes of reproduction, the predominant mode of epistasis was alleviating (positive), with multiple mutations reducing fitness less than expected from their individual effects. This epistasis was weaker, however, in sexual than in asexual organisms.

## Discussion and Conclusions

Our experiments show that rapidly changing environments can promote the evolution of sex, but at the same time, they call attention to some limitations of this theory. In particular, the parameter space that favored sex was quite limited, and the origin of sexual reproduction was more difficult than its maintenance. We also failed to observe a preponderance of aggravating (negative) epistasis, which is a key component of the mutational deterministic hypothesis [13], another well-known theory for the evolution of sex, thus adding to evidence against this hypothesis obtained in other systems [14-16]. Instead, our results suggest that an indirect benefit for sexual reproduction might arise from increased genomic modularity, perhaps leading to greater evolvability that sustains long-term

increases in fitness [17-19].

More generally, the studies summarized here highlight the utility of digital organisms for testing complex evolutionary theories because they allow one to manipulate any relevant features of the environment, control for the confounding effects of ancestry, compare the origin and maintenance of organismal traits under the same conditions, and obtain data across many replicate populations and for many thousands of generations. Finally, the insights gained from our experiments with digital organisms may also lead to future research on biological systems to examine the generality of these results.

**Acknowledgments.** The work presented here was supported by grants DEB-9981397, CCF-0643952, CCF-0523449 from the US National Science Foundation and DARPA 'Fun Bio' Program HR0011-05-1-0057.

## References

1. Weismann A. 1889. *Essays upon Heredity and Kindred Biological Problems*. Oxford: Clarendon Press.
2. Rice WR. 2002. Experimental tests of the adaptive significance of sexual recombination. *Nat Rev Genet* 3: 241-51.
3. Bell G. 1982. *The Masterpiece of Nature*. Berkeley: Univ. California Press.
4. McPhee CP, Robertson A. 1970. The effect of suppressing crossing-over on the response to selection in *Drosophila melanogaster*. *Genet Res* 16: 1-16.
5. Misevic D, Ofria C, Lenski RE. 2010. Experimental evidence for evolution of sex in changing environments *J Hered* 101: S46-S54.
6. Misevic D, Ofria C, Lenski RE. 2006. Sexual reproduction reshapes the genetic architecture of digital organisms. *Proc R Soc Lond B* 273: 457-64.
7. Lenski RE, Ofria C, Collier TC, Adami C. 1999. Genome complexity, robustness and genetic interactions in digital organisms. *Nature* 400: 661-4.
8. Wilke CO, Wang JL, Ofria C, Lenski RE, Adami C. 2001. Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature* 412: 331-3.
9. Misevic D, Ofria C, Lenski RE. 2004. Sexual reproduction and Muller's ratchet in digital organisms. In *Proceedings of Artificial Life IX*, ed. JB Pollack, et al, pp. 340-5: MIT Press, Cambridge, Massachusetts.
10. Ray TS. 1991. An approach to the synthesis of life. In *Artificial Life II*, ed. CG Langton, et al. Cambridge, Massachusetts: MIT Press.
11. Oros N, Nehaniv CL. 2009. Dude, where is my sex gene? — Persistence of sex over evolutionary time in cellular automata. *IEEE Symposium on Artificial Life*, Nashville, Tennessee.
12. Lenski RE. 1999. A distinction between the origin and maintenance of sex. *J Evol Biol* 12: 1034-5.
13. Kondrashov AS. 1982. Selection against harmful mutations in large sexual and asexual populations. *Genet Res* 40: 325-32.
14. Chao L. 1988. Evolution of sex in RNA viruses. *Journal of Theoretical Biology* 133: 99-112.
15. Elena SF, Lenski RE. 1997. Test of synergistic interactions among deleterious mutations in bacteria. *Nature* 390: 395-8.
16. Wilke CO, Lenski RE, Adami C. 2003. Compensatory mutations cause excess of antagonistic epistasis in RNA secondary structure folding. *BMC Evolutionary Biology* 3: 3.
17. Wagner GP, Altenberg L. 1996. Complex adaptations and the evolution of evolvability. *Evolution* 50: 967-76.
18. Earl DJ, Deem MW. 2004. Evolvability is a selectable trait. *Proc Natl Acad Sci USA* 101: 11531-6.
19. Woods RJ, Barrick JE, Cooper TF, Shrestha U, Kauth MR, Lenski RE. 2011. Second-order selection for evolvability in a large *Escherichia coli* population. *Science* 331: 1433-6.

# Modeling Cell Division of *B. subtilis* Using Dynamic Division of Reaction Spaces in a Membrane Artificial Chemistry

Takuma Miyake<sup>1</sup> and Kazuto Tominaga<sup>1</sup>

<sup>1</sup>Tokyo University of Technology  
tomi@acm.org

## Abstract

Localization of molecules in a natural cell plays important roles in interesting behavior of organisms like cell division and morphogenesis. Such localization is mostly formalized in a continuous space or lattice. This paper takes another approach using an artificial chemistry with membranes; we propose a dynamic division of reaction spaces to deal with molecular localization. As an application of the method, we modeled the cell division of *B. subtilis*. We executed the model on a simulator and observed the intended results.

## Introduction

Living organisms show many kinds of interesting behavior whose mechanisms are not easily understood. They include reproduction, morphogenesis, evolution, etc. In some of them, the properties and dynamics of lipid membranes play important roles. As one of the main interests in the field of artificial life is to understand the essence of living system, numerous formalisms have been proposed and used to model the behavior of life in which membranes take their part; artificial chemistries (AChems) are among them (Dittrich et al., 2001). For example, Madina et al. studied the formation of proto-cell structures using their 3D Lattice Artificial Chemistry (Madina et al., 2003). They observed in the model that amphiphilic molecules are organized into membrane-like structures.

Besides the properties of membranes, another factor is also important to understand interesting behavior: localization of molecules. For example, in the early stage of *C. elegans* (a kind of worm) embryogenesis, the point where the sperm enters decides the localization of specific proteins, which induces asymmetric cell division (Kemphues, 2000). It is beneficial for a formalism to be capable of dealing with such localization.

There seem to be two established methods to handle it: one assumes a continuous space and the positions of molecules; the other employs a lattice (1D, 2D or 3D, of squares or other shapes) and places molecules in lattice cells (Arjunan and Tomita, 2010). But both methods would have drawbacks when they are to be applied to modeling and simulating a life-like system with many compartments separated

by membranes. The first method may require much computational resource to calculate the behavior of molecules. With the second method, it seems difficult to scale and adapt the lattice size and granularity when, for example, morphogenesis from zygote to adult is to be modeled and simulated.

In this study, we take a different approach. Instead of using the position of molecule in a continuous space or introducing a pre-defined spatial structure, we divide reaction spaces dynamically. To express it, we extend our AChem (Amari and Tominaga, 2009). Then we model the cell division of *B. subtilis* to evaluate the expressiveness of the extended AChem.

The organization of the following sections is as follows. First, we illustrate part of the cell-dividing mechanism of *B. subtilis* which we are going to model. Second we briefly explain the base AChem and its extension. Then we model *B. subtilis* cell division and show the result of its execution. Finally, we discuss the proposed approach.

## Mechanism of *B. subtilis* Cell Division

*B. subtilis* is a gram-negative rod-shaped bacterium (Adams and Errington, 2009). It is a model organism in molecular biology. Its cell division has been studied, by which *B. subtilis* reproduces itself, for it is a single-cell creature. The mechanism of division is not completely understood, yet some details have been elucidated up until today.

This section illustrates part of the mechanism that controls the division of *B. subtilis* cell which we model in our AChem.

### Forming of Z-ring and division septum

In the process of division, a Z-ring and a division septum are formed at the mid-cell of *B. subtilis* (Adams and Errington, 2009; NW and J, 2005) (Fig. 1). A Z-ring is a ring-shaped polymer of cytoplasmic protein named *FtsZ*; it is formed by the polymerization of the protein on the inside surface of cytoplasmic membrane. Then the Z-ring constricts towards the deep-cell, and the septum formation follows it; the septum becomes one pole of each daughter cell when the division is complete.

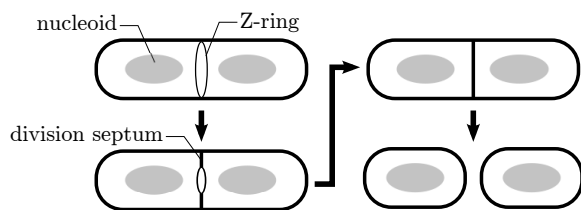


Figure 1: The Z-ring and septum.

In order for a cell to divide evenly, the position of Z-ring (and septum) must be regulated. Two mechanisms are regarded as contributing to the regulation, namely, *nucleoid occlusion* (Adams and Errington, 2009) and the *MinCDJ system* (Adams and Errington, 2009; van Baarle and Bramkamp, 2010; Bramkamp et al., 2008). Nucleoid occlusion prevents the Z-ring from forming near nucleoids (shown as gray ellipses in Fig. 1), while the MinCDJ system prevents one from forming near the cell-poles. In the present study, we model the latter mechanism.

Four kinds of proteins play their roles in the MinCDJ system, namely, *MinC*, *MinD*, *MinJ* and *DivIVA*. DivIVA localizes to the inner surface of cytoplasmic membrane at the cell-poles. It recruits MinJ, and MinJ recruits MinD, and MinD recruits MinC. MinC then prevents the polymerization of FtsZ near the cell-poles. Although the mechanism of the localization of DivIVA to the cell-poles is not yet fully understood, the protein is known to have a characteristic that tends to bind to a concave curve of lipid membrane surface (Ramamurthi and Losick, 2009; Lenarcic et al., 2009).

## Completion of cell division

These mechanisms restrict the Z-ring and the division septum to be formed at the mid-cell. The constriction of Z-ring makes the septum curve inward, so DivIVA binds to the cytoplasmic membrane near the Z-ring (Ramamurthi and Losick, 2009; Lenarcic et al., 2009). Then DivIVA recruits MinJ, MinD and MinC proteins, which will work again in the next cell division.

Following the completion of Z-ring constriction, the synthesis of division septum is complete, which is the end of cell division. The Z-ring at a new cell-pole is depolymerized by MinC and other proteins (Gregory et al., 2008); the FtsZ monomers will re-polymerize to form the next Z-ring.

## The Base Artificial Chemistry

The present study attempts to model the cell division of *B. subtilis* using an extended AChem, which we propose in this paper. Before we describe the extension, we give an outline of the base AChem (Amari and Tominaga, 2009).

A *v-atom* is an atom in this AChem, whose name starts with an upper-case letter followed by lower-case letters and/or digits, such as Abc and D2e. A *v-molecule* is a stack

of one or more lines of v-atoms. Shown in Fig. 2 is an example of v-molecule consisting of two lines, which is denoted by 0#AbCd/1#EfGh/, where 1 is the displacement of the second line relative to the first.

A *recombination rule* is a chemical equation in this AChem, which is phrased in terms of *patterns*. A pattern matches (or does not match) a v-molecule. A pattern consists of *atomic patterns* and/or *wildcards*.

An atomic pattern is denoted by a name of v-atom, and matches that v-atom; for example, the atomic pattern Ab matches a v-atom Ab.

There are two kinds of wildcards, namely, *atomic wildcard* and *sequence wildcard*. An atomic wildcard, denoted by a non-negative integer and surrounding angle brackets like <1>, matches any v-atom. The integer is the wildcard's ID, which is referred to by recombination. A sequence wildcard, denoted using an asterisk like <\*> or <3\*>, matches any sequence of zero or more v-atoms. The pattern shown in Fig. 3 (left) is denoted by 0#<\*>Ab<1><2\*>/0#Cd<3\*>/, and matches all of the three v-molecules shown in the right of the figure.

The left-hand side of recombination rule consists of one or two patterns, and the right-hand side may have any number of patterns. A recombination rule recombines a v-molecule (or v-molecules) matched by its left-hand side to v-molecule(s) represented by the pattern(s) on the right-hand side. Shown below is an example recombination rule.

$$0\#<*>Ab<1><2*>/ + 0\#Cd<3*>/ \\ \rightarrow 0\#<*>Ab<1><2*>/0\#Cd<3*>/ \quad (1)$$

If this rule is applied to the two v-molecules 0#ZyAbEf/ and 0#CdGhIj/, they are recombined to one v-molecule of the form 0#ZyAbEf/1#CdGhIj/.

In this AChem, a *membrane* surrounds a *cubicle*. Membranes can be nested to make a *system*. A system can model a natural cell including cell organelles. Each cubicle has a multiset of v-molecules, and so does each membrane; both are called *reaction spaces*. Each reaction space has its own set of recombination rules. Although reaction spaces are assigned to membranes and cubicles, each reaction space has no spatial structure; it is “well-stirred,” i.e., any v-molecule can react with any other v-molecule in the space. An example system is shown in Fig. 4(a). A system is represented by a tree structure, where a cubicle corresponds to a node and a membrane to an edge (Fig. 4(b)).

A v-molecule in the reaction space of a membrane (which can model a protein embedded in a lipid bilayer membrane),

Ab	Cd	
	Ef	Gh

Figure 2: An example v-molecule.



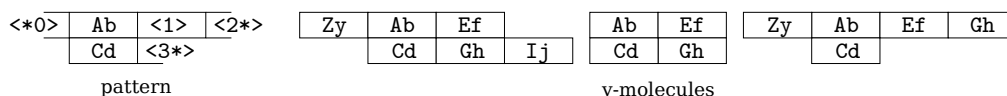


Figure 3: A pattern using sequence wildcards and its matching example v-molecules.

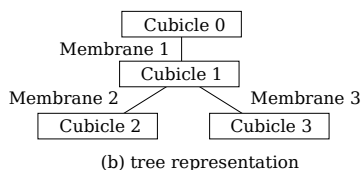
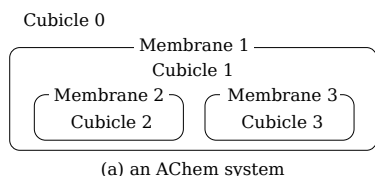


Figure 4: An example system of our AChem.

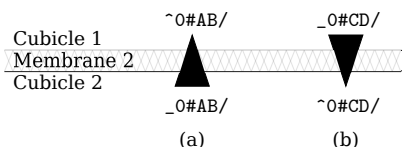
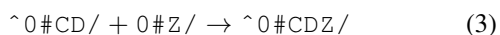
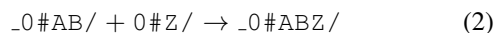


Figure 5: Directions of membrane v-molecules.

called *membrane v-molecule*, has its direction, as the membrane protein does. The direction of membrane v-molecule is *top* (represented by a preceding hat sign (^)) or *bottom* (by an underscore (\_)), and is relative to an adjacent cubicle from which the v-molecule is viewed. If a membrane v-molecule is top when it is viewed from a cubicle (as  $\sim 0\#AB/$  viewed from Cubicle 1 in Fig. 5(a) for example), it is bottom when viewed from the opposite cubicle ( $_0\#AB/$  from Cubicle 2).

A recombination rule specifies the types of v-molecules using directions. In a recombination rule of cubicle, if a pattern has no direction such as those in Rule (1), it represents a v-molecule in the reaction space of the cubicle. If a pattern has a preceding direction, as in the following examples, it represents a v-molecule having that direction in the reaction space of an adjacent membrane.



For example, if Rule (2) is applied to a bottom v-molecule  $_0\#AB/$  of a membrane (suppose the rule is defined in Cubicle 2 of Fig. 5 and we are viewing the v-molecule (a) from Cubicle 2) and a cubicle v-molecule  $0\#Z/$  (in Cubicle 2, not shown in the figure), they are recombined to a bottom mem-

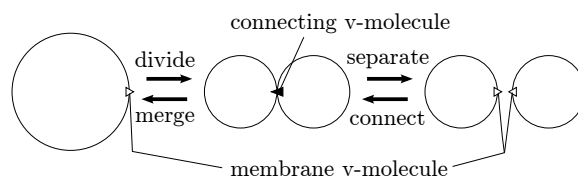


Figure 6: Membrane dynamics in this AChem.

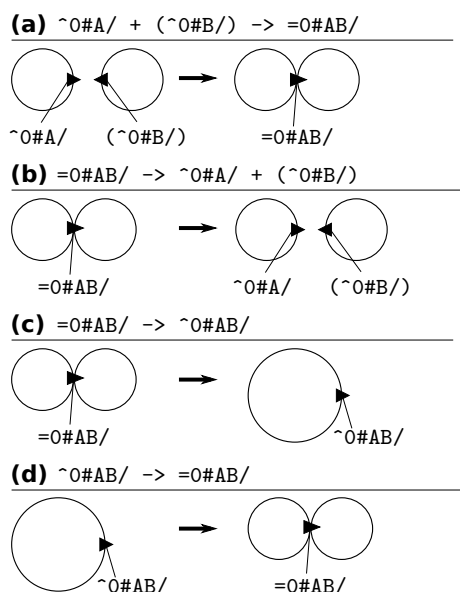


Figure 7: Recombination rules change membrane structure.

brane v-molecule (of Membrane 2) of the form  $0\#ABZ/$ , i.e.,  $_0\#ABZ/$ .

The AChem can express the division and merger of membranes (Fig. 6). The processes go through an intermediate state where two membranes are connected by a v-molecule. This v-molecule is called *connecting v-molecule*; it is represented by equal sign (=) in a pattern. The division and merger of membranes are not described by specifying membranes explicitly; instead, they are defined in terms of recombinations of v-molecules. Four kinds of recombination rules that change membrane structures are shown in Fig. 7 (the rules are supposed to be given to the parent cubicle of the membranes in this case). A pattern surrounded by parentheses like the second term of “ $\sim 0\#A/ + (\sim 0\#B/)$ ” expresses that the two patterns represent v-molecules of different membranes.

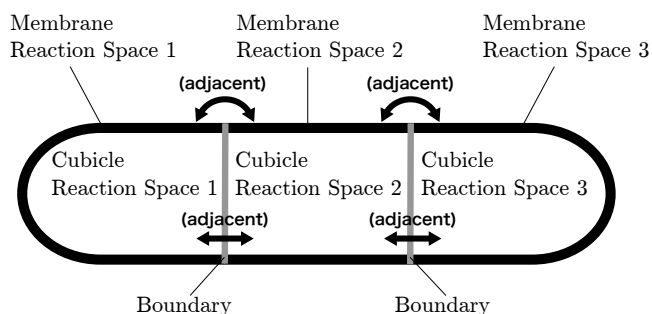


Figure 8: Membrane, cubicle and reaction spaces.

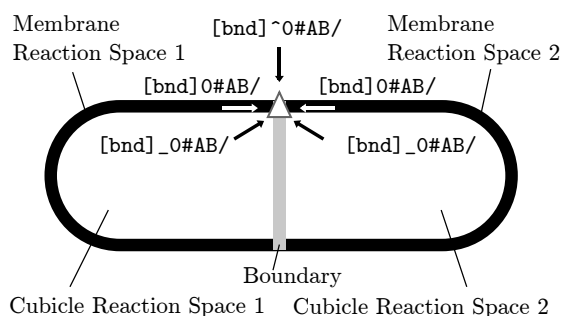


Figure 9: Reference to a boundary v-molecule.

A system is interpreted nondeterministically as follows.

1. Initialize the system: each reaction space is given initial v-molecules.
2. Choose a reaction space  $S$ .
3. Choose a recombination rule  $R$  from  $S$ .
4. Choose one or two v-molecules, if any, that  $R$  can apply to.
5. Recombine the v-molecule(s); change the membrane structure if specified.
6. Go to Step 2.

When a system is run on a simulator software, choices are made by a specific algorithm of the simulator (called its *reactor algorithm*). Some of our simulators make choices randomly; others employ physicochemical methods.

## Extensions to the Base Artificial Chemistry

This study extends the base AChem described in the previous section, and models the cell division of *B. subtilis* with the extended AChem. This section illustrates the extension.

In the process of the cell division, there occur the recruitment of division proteins (DivIVA, MinC, etc.) to the cell-poles and the localization of FtsZ at the mid-cell. Since the base AChem gives one reaction space to a cubicle and employs the well-stirred reactor algorithm, it cannot express such localization of proteins in a straightforward manner.

The present study extends the AChem so that a cubicle can have multiple reaction spaces, and so can a membrane, to express such localization. Reaction spaces of a cubicle (or a membrane) have adjacency relationships among them. Figure 8 depicts a cubicle that have three reaction spaces (Cubicle Reaction Spaces 1, 2 and 3), and its surrounding membrane that also have three reaction spaces (Membrane Reaction Spaces 1, 2 and 3); the arrows indicate their adjacency relationships.

## Boundary between reaction spaces

Two adjacent reaction spaces of a cubicle/membrane have a *boundary* between them. Unlike a membrane, a boundary has no reaction space. A boundary can be specified by a membrane v-molecule. This special kind of v-molecule is called *boundary v-molecule*. A boundary v-molecule has a direction. It can be viewed from the outside and the inside of the membrane (same as a normal membrane v-molecule), and also can be viewed from the reaction spaces it specifies. Figure 9 illustrates how a boundary v-molecule can be viewed from reaction spaces around it. The boundary separates the membrane into two reaction spaces (Membrane Reaction Spaces 1 and 2) and the cubicle into two (Cubicle Reaction Spaces 1 and 2). Each reaction space can refer to the boundary v-molecule in its recombination rules using the pattern shown near the arrow from the space; a boundary v-molecule is expressed by a tag “[bnd]” in a pattern.

## Migration of v-molecules between reaction spaces

In a natural cell, most of materials in cytoplasm can freely diffuse in the cytoplasm. To express such behavior, a recombination rule that makes v-molecules migrate between adjacent reaction spaces can be defined. This type of rule is called *migration rule*. An example rule is shown below:



The tag “[as]” means “another space.” When a rule of this type is applied to a v-molecule, the v-molecule migrates to any of the reaction spaces adjacent to the current space.

## Membrane division on a boundary

In the base AChem, a membrane is divided into two when a dividing rule is applied to a membrane v-molecule in the membrane. The division also divides the cubicle surrounded by the membrane; the contents of the cubicle (i.e., v-molecules and child membranes) are distributed to the new cubicles nondeterministically. This property, however, is not desirable when the AChem is to model cell division, because the contents of the cell should be divided evenly.

(a) [bnd]~O#AB/ -> =O#AB/

(b) [bnd]\_O#AB/ -> =\_O#AB/

(c) [bnd]\_O#AB/ -> =~O#AB/

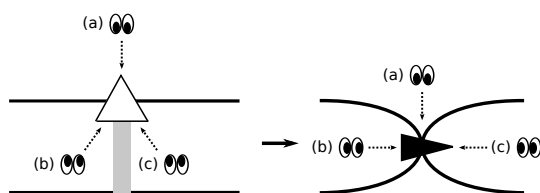


Figure 10: Membrane division on a boundary.

Dividing membrane and its inside cubicle on a specified boundary enables the system to distribute the contents of membrane/cubicle as intended. One can distribute the contents of a membrane/cubicle to its reaction spaces (by recombination rules) before division, then can divide the membrane/cubicle to make two membranes/cubicles.

Such division is performed when a specific type of recombination rule is applied to a boundary v-molecule. Three types of rules and their effects are depicted in Fig. 10. Each pair of eyeballs indicates the reaction space where the recombination rule is defined. An application of any of the rules (a), (b) or (c) divides the membrane/cubicle on the left to the two distinct membranes/cubicles on the right; the black triangle represents a connecting v-molecule. At the same time, the boundary disappears.

### Dividing a reaction space

A reaction space is divided dynamically by the application of recombination rule to a membrane v-molecule. There are two types of rules. One is a rule that creates a boundary molecule (Fig. 11(a)). An application of such a rule makes the membrane v-molecule a boundary v-molecule, divides the membrane reaction space where the membrane v-molecule has been residing, and also divides the inside cubicle reaction space from which the v-molecule can be viewed. The contents of each of the original reaction spaces are distributed nondeterministically to its daughter spaces.

The other is a rule that creates “neighboring space” (Fig. 11(b)), which is indicated by the tag “[nsp]”. When this type of rule is applied to a membrane v-molecule, a new cubicle reaction space that is adjacent only to the current (i.e., one having the rule) cubicle reaction space is created, and also a new membrane reaction space adjacent only to the current membrane reaction space (where the membrane v-molecule belongs to) is created. The contents of the original reaction spaces are distributed nondeterministically to the daughter spaces in the same manner as that for the previous case. In this type of division, the created boundary has no boundary v-molecule.

## Modeling the Cell Division of *B. subtilis*

Using the extended AChem, we constructed a model for the cell division of *B. subtilis*.

### Overview of the model

The conceptual diagram of the model is shown in Fig. 12. A small triangle represents a complex of MinC, MinD and MinJ, a small square represents DivIVA, and a small circle represents FtsZ. The process of division progresses as follows. (The numbers correspond to those in the figure.)

1. DivIVA-MinCDJ complex localizes to the inner surface of the both ends of the rod-shaped cell. FtsZ molecules are scattered over the whole cytoplasm.
2. FtsZ binds to any part of the inner surface of cytoplasmic membrane and starts to polymerize.
3. DivIVA-MinCDJ complex at the rod ends depolymerizes FtsZ polymers around it.
4. Since DivIVA-MinCDJ does not exist at the mid-cell, FtsZ polymerizes there.
5. The polymer of FtsZ becomes a Z-ring.
6. A septum starts to be synthesized as the Z-ring constricts. As the septum grows, DivIVA in the cytoplasm binds near the Z-ring.
7. The cell divides into two when the Z-ring constricts completely and the septum is fully synthesized.
8. MinC, MinD and MinJ binds to DivIVA that is recruited by the Z-ring, to make DivIVA-MinCDJ complex.
9. MinC in the complex depolymerizes the remaining FtsZ polymer. Go to Step 1.

### Definition of the model

The model is defined by the following specifications: the structure of the AChem system (membranes, cubicles and reaction spaces), the initial multiset of v-molecules for each reaction space, and the set of recombination rules for each reaction space.

The structure of the system is shown in Fig. 13. It consists of a membrane and a cubicle surrounded by the membrane. The membrane has three reaction spaces, namely, m-left-pole, m-mid-cell and m-right-pole; the cubicle also has three reaction spaces, left-pole, mid-cell and right-pole. All the membrane reaction spaces share the same set of four recombination rules; the cubicle reaction spaces also share the same set with each other, which comprises 13 rules.

Initial v-molecules are given as follows. The membrane reaction spaces at both ends of the cell, m-left-pole and m-right-pole, are given v-molecules representing DivIVA (\_O#Div4a/). The space m-mid-cell is given no v-molecule. All the cubicle reaction spaces, left-pole, mid-cell

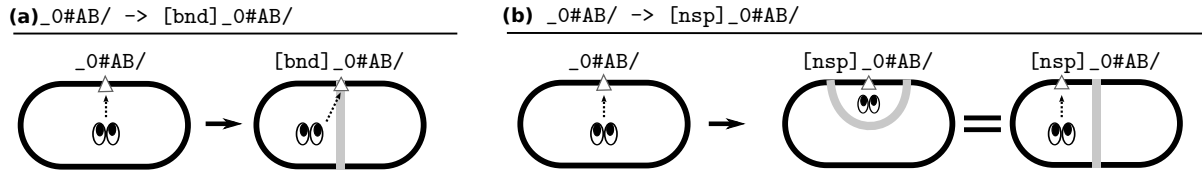


Figure 11: Examples of boundary creation.

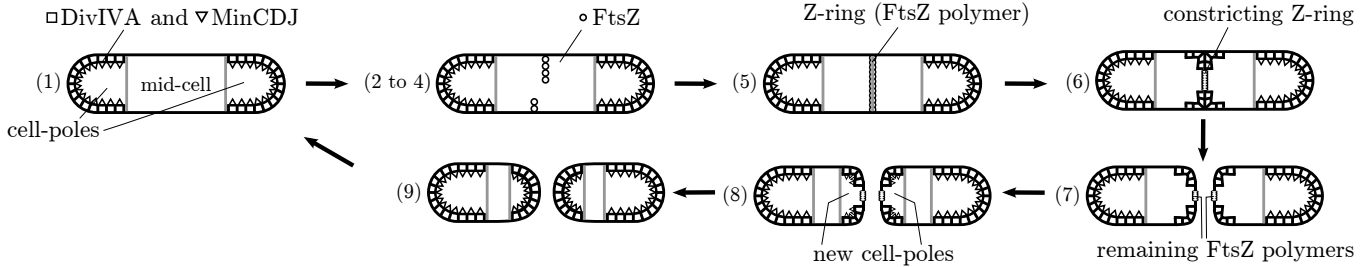


Figure 12: A conceptual diagram of the model.

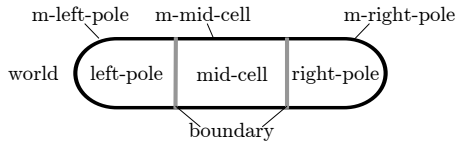
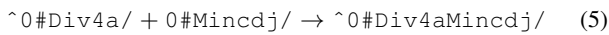


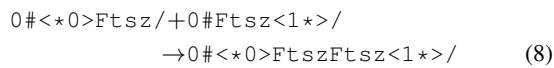
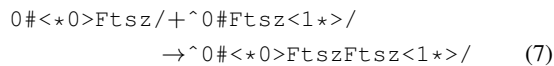
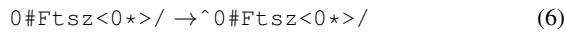
Figure 13: Initial structure of the system.

and right-pole, are given v-molecules for FtsZ,  $0\#Ftsz/$ , and v-molecules  $0\#Mincdj/$ , which represent MinC, MinD and MinJ at once. We deal with the three proteins as an abstract molecule to make the description short.

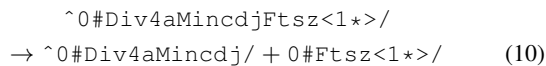
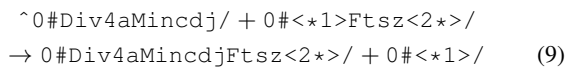
Recombination rules define how the system works as follows. First, MinCDJ binds to DivIVA (5).



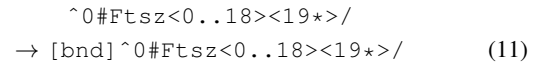
And FtsZ binds to the plasmic membrane (6). To the protein, an FtsZ monomer binds to polymerize (7); FtsZ polymers in the membrane also join (8).



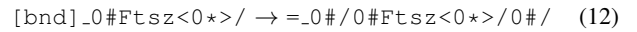
MinC in DivIVA-MinCDJ complex binds to FtsZ polymer (9) and depolymerizes it (10).



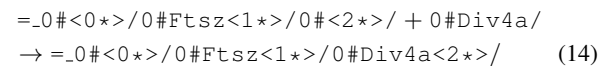
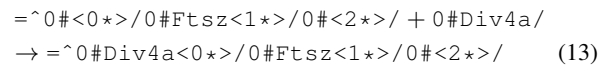
Because FtsZ polymerization is prevented at the ends of cell by DivIVA-MinCDJ complex, FtsZ polymerizes at the mid-cell. Eventually a Z-ring is formed (11) (we regard a polymer of twenty FtsZ as a Z-ring; the number is arbitrary). The Z-ring is represented by a boundary v-molecule. The rule divides each of m-mid-cell and mid-cell into two reaction spaces.



The Z-ring constricts to the deep-cell and divides the membrane (12); the constricted Z-ring becomes a connecting v-molecule.



DivIVA in cytoplasm binds to the plasmic membrane near the constricted Z-ring (13, 14). Note that the viewpoints of the two rules are in the opposite cubicle to each other. DivIVA near the Z-ring is expressed by a v-molecule with multiple lines.



The divided membranes separate when a particular number (ten) of DivIVA have bound to the both sides of Z-ring (15), meaning sufficient time has elapsed for the complete constriction of Z-ring. (The first term is folded to fit in the

column.)

```
=0#<0..9><10*>/0#<*11><12..31><32*>/
      0#<33..42><43*>/
→ _0#<0..9><10*>/0#<*11><12..21>/
+ (_0#<33..42><43*>/0#<22..31><32*>/) (15)
```

After the division, there are remaining FtsZ and DivIVA binding nearby at the new cell-pole. In other words, they indicate the new cell-pole. The following rule creates a new reaction space that represents the new cell-pole (16).

```
^0#<0..9><10*>/0#Ftsz<11*>/
→ [nsp]^0#<0>/ + [nsp]^0#<1>/ + [nsp]^0#<2>/
+ [nsp]^0#<3>/ + [nsp]^0#<4>/ + [nsp]^0#<5>/
+ [nsp]^0#<6>/ + [nsp]^0#<7>/ + [nsp]^0#<8>/
+ [nsp]^0#<9>/ + [nsp]^0#<10*>/
+ [nsp]^0#Ftsz<11*>/ (16)
```

Then MinCDJ binds to DivIVA at the new cell-pole to make DivIVA-MinCDJ complex (5), and MinC there depolymerizes the FtsZ polymer remaining at the pole (6).

In addition, the system has migration rules (like (17)) that let v-molecules migrate to adjacent cubicle reaction spaces, and a rule that expresses the decomposition of DivIVA (18).

```
0#Div4a/ → [as]0#Div4a/ (17)
0#Div4aDiv4a<0*>/ → 0#Div4a/ + 0#Div4a<0*>/ (18)
```

## Execution of the model

We built a prototype simulator for the extended AChem by modifying a simulator for the base AChem; the both simulators are written in Ruby. When we ran the description for the cell division illustrated in the previous section, the model worked as intended.

A snapshot taken from the execution is shown in Figure 14. The text above is the output of simulator, and it is depicted in the illustration below. In this state, the first cell division is complete, and each daughter cell has started the next cycle of cell division. In Membrane 11 and Membrane 13, remaining FtsZ polymers (`_0#FtszFtsz...`) are observed.

## Discussion

In the extended AChem, while each reaction space is well-stirred, a cubicle/membrane can consist of multiple reaction spaces, so localization of molecules within the cubicle/membrane can be expressed.

The division of reaction spaces is performed by the application of recombination rule to a molecule. This is in the same framework we used to formalize membrane division and merger (Tominaga et al., 2007). The main advantage of this approach is that the same set of rules can be applied

after the structure of system has changed because rules do not refer to membranes, cubicles or reaction spaces by their IDs, positions, coordinates or addresses; the behavior is determined only by v-molecules they have. Though we did not model nucleoid occlusion, we think it can also be modeled using the division of reaction spaces.

Possible topologies of reaction spaces are limited. For example, a membrane reaction space and the cubicle reaction space inside (and adjacent to) it always correspond in a one-to-one manner.

In the illustrated application, the execution of the system is somewhat like a reaction-diffusion system; the number of FtsZ in the mid-cell space (or “concentration”) seems to contribute to the formation of the Z-ring. This is because the implementation of the simulator uses random numbers to decide which reaction to occur. The current implementation does not take physicochemical dynamics of molecules into account. Doing it will be our future work.

In (Madina et al., 2003), the formation of membrane-like structures is studied. They define the lattice and interaction among particles in the space. A membrane-like structure is observed as a collection of particles in lattice cells that enclose an area. So the lattice should be suitable for the study. In our AChem, a membrane is a primary entity and cannot be decomposed into parts; this property will be beneficial in modeling the behavior of membrane at a high level of abstraction.

A work using E-Cell to simulate the E-ring formation of *E. coli* (Arjunan and Tomita, 2010) predefines a hexagonal lattice with voxels having 12 neighbors in order to simulate the behavior of proteins in cytoplasm. In contrast, our study first only gives three reaction spaces to express the areas of cytoplasm and they divide dynamically as the execution progresses. This flexibility will contribute to the scalability of model, especially the membrane structure of which changes considerably, like the process of complete ontogenesis.

In this aspect, our approach has similarities with L-systems (Lindenmayer, 1968): symbols in an L-system can increase as rules are applied, and rules specify no position or ID of each symbol occurrence. Since ours is an artificial chemistry, membranes and cubicles can have (v-)molecules, and reactions among them can be described as recombination rules. We think this is advantageous in modeling a system based on known biochemical reactions.

## Concluding Remarks

In this paper, we presented a membrane artificial chemistry that can dynamically divide reaction spaces, as an extension to our previous artificial chemistry. The extension is introduced to express the localization of molecules.

We showed an application of it: a model for the cell division of *B. subtilis*. The model is defined by the initial structure, the initial v-molecules and 17 recombination rules. We executed the model on our simulator, and observed that a



# A Cultural Evolutionary Model for Artifact Capabilities

Felicitas Mokom<sup>1</sup> and Ziad Kobti<sup>1</sup>

<sup>1</sup>University of Windsor, Windsor, ON Canada N9B-3P4  
mokom@uwindsor.ca and kobti@uwindsor.ca

## Abstract

The use of tools or artifacts is essential to the human race and has been the subject of recent research in Artificial Intelligence. How individual agents acquire these capabilities and how they evolve can be considered vital steps towards understanding complex group capabilities. In a previous study, we designed and implemented an extended version of a theoretical model for artifact capability that accommodated biological evolution and learning via exploratory methods. Historical knowledge and genetic algorithms were combined with learning techniques to build agents that could learn either individually from observations of their own behaviour or socially by observation from a distance. In this study, we incorporate a collaborative form of cultural learning into the model in an effort to enhance the artifact capability-learning agents. This is accomplished via the design of a cultural evolutionary model that utilizes genetic and cultural algorithms to complement the cognitive abilities of the agents. Learning agents belonging to a social network cooperate with and benefit from each other by sharing individual experiences. Results obtained from the multi-agent simulation implementation confirm the efficiency of social learning over individual learning and demonstrate the benefits of cultural over biological evolution. They also suggest that as artifacts get more complex, social agents learning via cultural influence outperform those learning by observation from a distance.

## Introduction

The ability of humans to learn tool or artifact use, evolve these capabilities and transfer the knowledge to others has been of much interest to various researchers particularly in the cognitive sciences. Archaeologists (Plummer, 2004) are fascinated by the earliest recordings of tool use, philosophers (Preston, 1998) theorize on the importance of tool use relative to human intelligence and behavioural geneticists (Bacher et al., 2010) present arguments on the role of genetics in tool use behaviour. Preston contends in her work that the study of tool use be considered as important as the study of language because it is indicative of the high level cognition that humans are capable of. According to (Petroski, 1992) artifact evolution is driven by functionality rather than failure. Artifacts do not necessarily evolve because they fail at what they were intended for, but rather because they can

always be improved. These improvements are often identified during use of the artifact. Humans use tools by themselves but often combine their tool capabilities. In order to successfully model these complex group capabilities it is essential to understand how humans acquire individual capabilities and how these capabilities change over time.

In this study the terms tools and artifacts are used interchangeably and include any physical object in the environment that a human agent can use towards achieving a goal. The human agent is a rational agent that acts in its best interests, has beliefs about the world, and chooses its actions accordingly (Wooldridge, 2000). Based on the Belief-Desire-Intention (BDI) theory of (Bratman, 1987) the rational agent has beliefs, desires and intentions. The agent's beliefs describe its informative state about the world. Its desires represent what the agent would like to accomplish and are used to devise goals. Its intentions are adopted goals that the agent uses to generate plans or actions that it performs. According to (Acay et al., 2008) tool capability resides within the intentions of an agent and represents plans that the agent can realize with the help of a tool. If an agent has capability for a tool then it has at least one plan that specifies one way to use the tool towards one or more of its adopted goals.

In a previous study (Mokom and Kobti, 2011) we implemented an extended version of Acay *et al.*'s theoretical model for tool capability incorporating biological evolution and learning through exploratory methods. A representation of artifacts and the cognition of an agent that can learn artifact capabilities were provided. Learning techniques from (Russell and Norvig, 1995) were combined with genetic algorithms (GA) to build a multi-agent simulation that evaluated individual and social learning in the form of observational learning from a distance. The social learning agent observed another agent successfully apply an artifact capability without the acting agents' knowledge, noted partial information and subsequently formed a learning goal to apply the same capability.

One limitation of the previously implemented social learning agent is the fact that there must exist another agent in its vicinity that already possesses the capability to use



the tool. This limitation coupled with the contention by (Reynolds, 1997) that cultural evolution evolves faster than biological evolution is the inspiration for the work in this paper. We design and implement a cultural evolutionary model that supports agents that can socially learn an artifact capability without any prior knowledge. This is accomplished via the integration of a GA and a cultural algorithm (CA) with the framework of an artifact-capability learning agent. A learning agent can benefit from being part of a social network where individual experiences are shared by using the experiences of others to enhance its own learning process. Our objectives are to demonstrate how agents can collaboratively learn an artifact capability over time and compare the results to those obtained for observational learning agents.

The next section provides some background on related work. It is followed by our architecture of artifact capability-learning agents. We then provide details on our implementation and experiments conducted, followed by conclusions deduced and future work.

## Background

### Artifact Use

The subject of tool use particularly in animals has been explored in various fields. (Wood et al., 2005) provide a good background on this. Much of the underlying work involves the effort to understand how animals explore objects. (Power, 2000) provides some insight into exploratory methods utilized by children and animals when they encounter a new tool. He contends that the exhibited behaviour, which can sometimes be genetically predetermined, is species dependent and very much influenced by culture.

Robotic researchers have also explored the subject of tool use. This has involved the development of object recognition mechanisms in robots (Wood et al., 2005) and the creation of industrial robots programmed for specific tool use (Bluethmann et al., 2003). In an effort to investigate robots learning tool use through exploratory methods, (Stoytchev, 2005) provides a representation of a robot that can attempt various actions with a tool, record and remember the effects. (Schäfer and Bergfeldt, 2007) investigate the emergence of complex tool use behaviours acknowledging that they need to combine their efforts with learning and reasoning by agents in order to obtain more useful results. (Noble and Franks, 2002) simulate various social learning methods for tool use concluding that emulation is sometimes a more effective method of learning than imitation because it promotes exploration. Omitted from their research is an evolutionary aspect to their work.

### Cultural Learning

Knowledge among humans is often transmitted through experience and cooperation. According to Tomasello et al. (1993) in cultural learning, integrated patterns of behavior accumulate changes across generations of a social group.

They identify three different manifestations of cultural learning namely imitation, instructed learning and learning by collaboration. Cultural evolution describes the change of culture over time and can be used to study the effects of cultural learning.

(Curran and O’Riordan, 2007) simulate the instructed learning form of cultural learning using a teacher/pupil environment. In their study a GA and a neural network are used to evolve a population where fitter individuals are selected as teachers for the pupils of the next generation. The goal was to demonstrate how cultural learning improves the fitness of a population. (Acerbi and Nolfi, 2007) utilize simulated annealing to incorporate cultural evolution in their comparisons between individual learning and the imitation form of cultural learning concluding that a sequence of both yields the best results. Geared more towards robotics, much of their work involves robotic sensors and body schema. (Reynolds and Peng, 2004) capitalize on the emergence of cultural learning in their CA framework to demonstrate the power of learning and adaptation within cultures.

## Architecture of an Artifact Capability-Learning Agent

An artifact capability-learning agent has the ability to employ learning techniques towards acquiring a tool capability, that is one way to use a tool towards the achievement of one or more of its goals. This can be accomplished via individual or social learning experiences. In the former case the agent learns solely through observations of its own behaviour and in the latter, the agent learns by observing or co-operating with other agents in its environment. We present the existing model of an artifact capability-learning agent from our previous study and demonstrate its expansion to cultural learning agents.

### Cognitive Elements of Learning Agent

The cognition of a rational agent endowed with the ability to learn artifact capabilities, was incorporated into a general model of learning agents developed by (Russell and Norvig, 1995). This can be appreciated in Figure 1 obtained from (Mokom and Kobti, 2011). The learning agent’s cognition is composed of a performance element, a critic and a learning element. The performance element bears the responsibility of selecting the agent’s external actions to perform. Once these actions are performed, the critic element measures resulting percepts against an external predefined standard of performance and generates feedback. This feedback is received by the learning element and used to improve the performance element so it can do better the next time. The rational agent’s beliefs, goals and capabilities reside within the performance element playing a role in the decision making process of action selection.

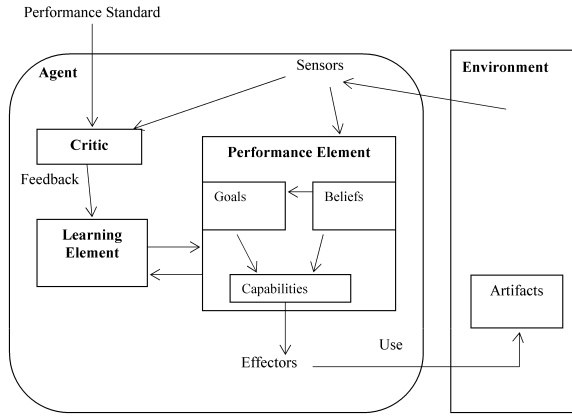


Figure 1: An artifact capability-learning agent

## Artifact and Agent Model

An artifact is represented as an object made up of one or more parts. Each artifact part is composed of a set of attributes. An artifact attribute has a set of possible values and a visibility property. The visibility property indicates whether an observing agent can copy the value of the attribute chosen by the agent it is observing. Let's consider a *pen* as an artifact. The part *shell* could represent the entire outer layer of the pen with an attribute *hold-position*. The set of values for the hold-position attribute indicate all possible points where the pen can be held. If the hold-position attribute is visible then an observing agent can copy the point at which the pen is held by the acting agent.

An artifact capability-learning agent has beliefs, goals and capabilities. A capability has an abstract functional ability and an ordered list of tasks. Abstract functional abilities represent all the things that an agent can do with an artifact regardless of whether the agent knows how. It is only when the agent acquires the knowledge to use the abstract ability that the agent can be described as having the capability. An agent can therefore select an abstract ability and use it to formulate a learning goal. For simplicity it is assumed in this study that an artifact has a single part and multiple attributes. The ordered lists of tasks represent attribute value sequences that the agent must determine in order to realize the capability.

External to the rational agent is a predefined standard of performance that is goal dependent for every artifact. This standard maintains information about the number of required tasks and the correct attribute value sequences for each task within an artifact capability. The performance standard is used by the critic element in evaluating the results of the agent's actions. The critic's feedback includes an average fitness score for the attempted sequence. The model supports a *range of values* performance standard that provides an inclusive range within which the selected attribute value is constrained to fall. For an attribute value sequence

$V = \langle v_1, \dots, v_n \rangle$  with  $n$  attributes, the critic calculates the mean fitness score  $MF$  as follows:

$$f(v_i) = \begin{cases} 1, & mn \leq v_i \leq mx \\ \frac{1}{mn-v_i}, & v_i < mn \\ \frac{1}{v_i-mx}, & v_i > mx \end{cases}$$

$$MF(V) = avg \left( \sum_{i=1}^n f(v_i) \right)$$

where  $mn$  represents the lower bound of the performance standard for the attribute with value  $v_i$  and  $mx$  represents its upper bound. The function gives the same score to all values that satisfy the standards' criteria. The rest of the values are scored based on their distance from the required range.

## Performance Element

One of the key decisions in the design of a learning agent is the design of the performance element. In accordance with (Russell and Norvig, 1995), the performance element of an artifact capability-learning agent should contain all the information needed by the agent to go about trying to use the tool. This is essentially how the agent deliberates and selects attribute values. We inherit two types of performance elements designed in the previous study and refer to them henceforth as PE1 and PE2. In this study we design a new type of performance element PE3.

All three performance elements' maintain a history of failed attempts in their respective beliefs. A *fitness-based attribute value selection* procedure is used in the selection of attribute values where one randomly chosen attribute of a selected sequence is modified at each attempt. For PE1 and PE2 the selection is based on the fitness of the agent's previous attempts. PE1 supports an agent learning on its own. The agent simply ensures that it does not repeat attribute value sequences that have previously failed. PE2 supports an agent learning socially via observation from a distance. Like PE1, it does not repeat attempted sequences. The variation lies in the fact that PE2 has partial knowledge of the capability at the start of the learning process. In determining new attribute values, the agent only selects and modifies the invisible attributes.

The new performance element PE3 is built to support an agent learning a tool capability through cultural experience. Its selections are based on both the fitness of its previous attempts and the fitness of all other agents that it cooperates with.

## Social Network and Cultural Algorithm

PE3 agents that collaborate with each other belong to a social network. In this study, it is not deemed necessary to define a social network with complex relationships. The social

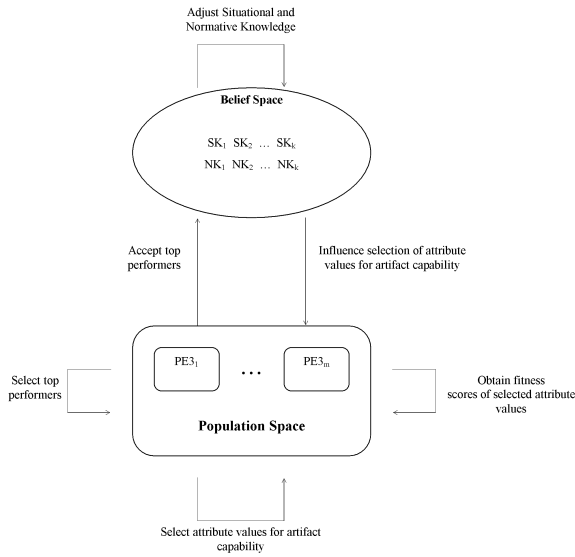


Figure 2: Cultural learning by  $m$  agents of a  $k$ -task artifact capability

network exists only to facilitate the exchange of information between agents towards enhancing the learning process.

PE3 agents are designed within the CA framework. CA's were introduced by (Reynolds, 1979) to facilitate the modeling of cultural evolution. A CA is made up of a belief space, a population space and a communication protocol between them. Selected individuals from the population space contribute to knowledge maintained in the belief space. The contribution is transmitted through an acceptance function and the knowledge in the belief space is adjusted accordingly. That knowledge influences the evolution of the individuals in the population space via an influence function. A CA supports the use of any kind of evolutionary algorithm in the implementation of the population space. The framework for PE3 agents learning an artifact capability is shown in Figure 2. It demonstrates PE3 agents belonging to a single social network sharing one global belief space within the CA. In the figure, there are  $m$  agents trying to learn the same artifact capability with  $k$  tasks.

**Knowledge Sources** Reynolds identifies five types of cultural knowledge that can be maintained in the belief space of a CA. They are situational, normative, topographic, historical or temporal and domain knowledge. Figure 2 shows the belief space in our design using situational and normative knowledge. Situational knowledge maintains the best performers so far. For artifact capability-learning agents cooperating with each other, these would be the highest scoring selections of attribute value sequences. Normative knowledge maintains encouraging ranges for each attribute value making it feasible for agents to “jump into the good range”

(Chung and Reynolds, 1998). The learning agents can adjust their attribute value selections using the guidance of these ranges that have been derived from selections of the top performers.

PE3 agents utilize knowledge from two types of belief spaces. The agent's personal belief space  $PB$  maintains a history of its failed attempts for the current task being learned and is local to the agent. Thus  $PB = \{\langle v_1, \dots, v_n \rangle\}$  where  $n$  represents the number of attributes for the artifact and each  $v_i$  is the selected attribute value for the sequence. The global belief space shared by agents in a social network henceforth referred to as  $GB$  is defined as:  $GB = \langle S, N \rangle$ , where  $S = \{SK_1, \dots, SK_k\}$  represents the situational knowledge and  $N = \{NK_1, \dots, NK_k\}$  represents the normative knowledge for  $k$  tasks of an artifact capability. The situational knowledge maintains the single best exemplar found so far for each task:

$$SK = (t, s_{SK}, \langle kv_1, \dots, kv_n \rangle) \quad (1)$$

where  $t$  is the task id,  $n$  represents the number of attributes for the artifact, each  $kv_i$  is the selected attribute value and  $s_{SK}$  represents the score of the sequence. The normative knowledge keeps favourable ranges for each attribute value. This is defined as:

$$NK = \{t, R_1, \dots, R_n\} \quad (2)$$

where  $t$  is the task id and  $n$  is the number of attributes for the artifact. Each  $R_i$  is a tuple:

$$R_i = \langle sl, su, [l, u] \rangle \quad (3)$$

where  $l$  and  $u$  represent the favourable lower and upper bound values of attribute  $i$ , with  $sl$  and  $su$  as their respective scores. A PE3 agent contributes to and uses both belief spaces in the learning process.

**Adjusting the Belief Spaces** The agent's local belief space is updated with the failed attempt every time the agent tries a new attribute value sequence for a particular task and fails. For  $GB$ 's adjustment when top performers are accepted, they are sorted according to their scores. If  $h$  contains parameters for the individual with the highest score:  $h = (t, s_h, \langle v_1, \dots, v_n \rangle)$ , then it is used to adjust the situational knowledge  $SK$  defined in Eq. (1) as follows:

$$SK' = \begin{cases} h, & s_h > s_{SK} \\ SK, & \text{otherwise} \end{cases} \quad (4)$$

Thus the situational knowledge is always the highest performer so far among all the attempted attribute values for the specific task by members of its social network.

In order to adjust the normative knowledge we need to deal with one attribute at a time. For each attribute  $i$  we

obtain and sort its values for all top performers. The lowest selected value  $x_i$  and the highest selected value  $y_i$ , with their corresponding scores  $sx_i$  and  $sy_i$  can now easily be extracted. Normative knowledge for the task being learned defined in Eqs. (2) and (3) is updated for each attribute  $i$  using the following formulae:

$$\begin{aligned} l'_i &= \begin{cases} x_i, & (x_i < l_i \text{ and } sx_i = sl_i) \text{ or } sx_i > sl_i \\ l_i, & \text{otherwise} \end{cases} \\ sl'_i &= \begin{cases} sx_i, & (x_i < l_i \text{ and } sx_i = sl_i) \text{ or } sx_i > sl_i \\ sl_i, & \text{otherwise} \end{cases} \\ u'_i &= \begin{cases} y_i, & (y_i > u_i \text{ and } sy_i = su_i) \text{ or } sy_i > su_i \\ u_i, & \text{otherwise} \end{cases} \\ su'_i &= \begin{cases} sy_i, & (y_i > u_i \text{ and } sy_i = su_i) \text{ or } sy_i > su_i \\ su_i, & \text{otherwise} \end{cases} \end{aligned} \quad (5)$$

Using these rules, the agents will progress towards learning the correct range required by the performance standard.

**Population Space and Influence from Global Belief Space** The population space in our cultural algorithm uses a genetic algorithm. As in the previous study the GA uses a bit representation for solutions in the population. It employs two-point crossover and mutation to modify a single attribute value for each attempt. Selection for reproduction is accomplished via roulette wheel selection. With PE3 agents however, mutation is carried out differently. In order to benefit from knowledge in *GB*, the situational and normative knowledge are used to determine direction and step size for the mutation respectively. This effectively permits attribute value sequences to follow the exemplar and at the same time strive to get into a desirable range. If the sequence being influenced is  $q = (s, \langle v_1, \dots, v_n \rangle)$ , then the chosen attribute's value  $v_i$ , is mutated using the following formula derived from Chung and Reynolds (Chung and Reynolds, 1998):

$$v'_i = \begin{cases} v_i + |(u_i - l_i) \cdot N(0, 1)|, & v_i < kv_i \\ v_i - |(u_i - l_i) \cdot N(0, 1)|, & v_i > kv_i \\ v_i + (u_i - l_i) \cdot N(0, 1), & \text{otherwise} \end{cases} \quad (6)$$

where  $kv_i$  represents the exemplar value in the situational knowledge as defined in Eq. (1),  $l_i$  and  $u_i$  correspond to the lower and upper bounds for that attribute in the normative knowledge defined in Eqs. (2) and (3), and  $N(0, 1)$  is a random value obtained using the standard normal distribution. All values correspond to the current task being learned.

## Cultural Learning Simulation

The simulation environment is a simple 20 x 15 toroidal grid world, in which each square contains an agent and an arti-

fact. There are three types of agents henceforth referred to as AG\_GA\_PE1, AG\_SOCIAL\_PE2 and AG\_SOCIAL\_PE3 varying based on the implementation of the performance element. All AG\_SOCIAL\_PE3 agents belong to a single social network. The agents can learn capability for artifacts with different complexity. All artifacts are made up of a single part but differ in the number of attributes. The grid is populated with 100 members of each type of agent and the same type of artifact is placed in each square. The agents simultaneously learn the same artifact capability by attempting different combinations of attribute values employing the respective technique of their performance elements.

For the genetic algorithm of AG\_GA\_PE1 and AG\_SOCIAL\_PE2 a mutation rate of 0.01 was chosen. For AG\_SOCIAL\_PE3 agents, mutation was determined by direction and step-size with a mutation rate of  $1/n$  where  $n$  represented the number of attributes for the artifact being learned. The crossover rate was set to 0.7 and the population size at 100 for the GAs of all agents. The range of possible attribute values for artifacts was set to  $[1..100]$  with the *range of values* performance standard covering 20% of the range. The number of tasks required by all agents to learn to use the artifact was 5. Finally, the top 5% performers for each agent's solutions were selected to be accepted into the global belief space.

The pseudo-code for cultural learning of an artifact capability is shown as Algorithm 1. At the start the social network is created and its global belief space is initialized to 5 tasks. For each task the exemplar is set to null and the normative range to the range of possible attribute values that is  $[1..100]$ . Agents are then added to the network. All agents perform the rest of the algorithm simultaneously. Each agent gets the artifact at its location and uses its cognitive elements to learn the capability. The learning element selects an appropriate ability and formulates a goal. The performance element initializes the agent's local belief and capability to null and the goal to false. Every simulation step the agent generates an attribute value sequence, attempts it and performs the necessary updates. If the feedback of the attempt indicates failure, the sequence is added to the agent's local belief. If there is some form of success, the agent has either reached its goal or has met the minimum requirement to proceed to the next task. If the goal is achieved, the learning element advises the performance element to perform the final capability update and the agent is done. If the goal is not yet reached, the performance element is asked to update the capability with the learned task's successful attribute values, clear the local belief as well as the agent's population space and continue on to the next task.

The knowledge maintained by *GB* is utilized when PE3 provides an attribute value sequence. The pseudo-code is shown as Algorithm 2. POP\_SIZE is a constant that specifies the number of attribute value sequences being evolved by the GA used to implement PE3's population space and

---

**Algorithm 1** Cultural learning of an artifact capability

---

```
Create social network
Initialize global belief space
Add agents to social network
Each agent gets artifact
Learning element selects an ability
Learning element formulates a goal
PE3 initializes local belief
PE3 initializes goal and capability
while goal not achieved

    PE3 provides attribute value sequence
    Critic tests attribute value sequence
    Critic generates feedback
    Learning element generates changes
    PE3 applies changes
end
```

---

is set to 100 in our experiments. The initial population is randomly generated without repeating sequences that have been attempted already. Once that is complete PE3 provides attribute value sequences by checking if there are still sequences to be attempted, selecting and returning one. If all attribute value sequences in the population have been attempted, fitness scores are used as the criteria to vote top performers for acceptance into *GB*, which is responsible for its own adjustment. A new population is then generated influenced by *GB*'s situational and normative knowledge. One attribute value sequence is selected from the new population and returned for the agent to attempt.

CA's require the evaluation of the entire population space prior to communication with *GB* via the acceptance function. For an artifact capability-learning agent this means that an agent's selections have no impact on *GB* until after all attribute value sequences in the population have been attempted. This is necessary because the agent has to test every generated sequence and obtain its fitness before the top performers can be identified. There is one more instance however, where it would be useful to update *GB*'s knowledge. That would be when the critic element declares success for an agent either at the task or the goal level. This can occur at any time during the evaluation process of the population space. In this specific case, when applying the learning elements suggested changes PE3 requests that the successful attribute value sequence be accepted into *GB*. It does not vote for top performers since it is likely that the entire population has not yet been evaluated. After the successful sequence is accepted *GB* adjusts its situational and normative knowledge using the same rules as when it receives top performers. This allows the agents to benefit from the success of others.

## Experiments and Results

AG\_GA\_PE1 agents are the individual experience-learning agents that strive to acquire an artifact capability on their

---

**Algorithm 2** PE3's algorithm to provide attribute value sequence

---

```
if size(POP) < POP_SIZE

    values = Generate random value sequence
else
    if attempted all sequences in POP
        Select top performers from POP
        Accept selected performers in GB
        Generate POP' with influence from GB
        values = One value sequence from POP'
    else

        values = One value sequence from POP
    end
end
```

---

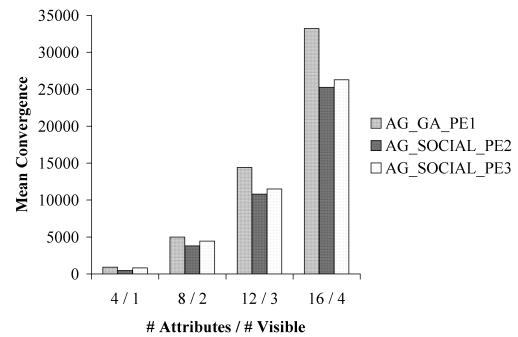


Figure 3: Average Convergence For All Agents Learning Capability for 4, 8, 12 and 16-attribute Artifacts (Visibility of attributes applies only to AG\_SOCIAL\_PE2 agents)

own. An AG\_SOCIAL\_PE2 agent learns socially by observing a capable agent from a distance, copying visible attributes and learning the remaining attribute values on its own. AG\_SOCIAL\_PE3 agents benefit culturally by cooperating with other agents in an effort to enhance their individual learning abilities. Figure 3 shows the results of 100 representatives of each type of agent learning capability for artifacts with 4, 8, 12 and 16 attributes. Figure 4 shows the results of 100 representatives of both types of social agents learning capability for artifacts with 8, 12, 16, 20 and 24 attributes. In all experiments 25% of the attributes were made visible for AG\_SOCIAL\_PE2. At the end of each test run, the mean convergence times for each type of agent were computed. These are the average number of iterations needed by the agents to learn the artifact capability.

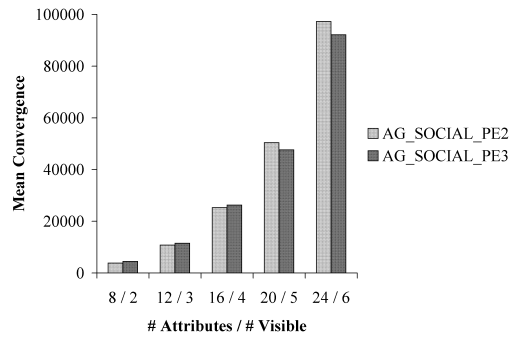


Figure 4: Average Convergence For Social Agents Learning Capability for 8, 12, 16, 20 and 24-attribute Artifacts (Visibility of attributes applies only to AG\_SOCIAL\_PE2 agents)

It can be observed in Figure 3 that AG\_GA\_PE1 agents were outperformed by both AG\_SOCIAL\_PE2 and AG\_SOCIAL\_PE3 agents in all conducted experiments. The results show an increase in the difference in convergence rate between the individual and social learning agents as the number of attributes increased from 4 to 16 attributes with the individual learning agents needing more time to learn the capability. An interesting observation in Figure 4 is the difference in convergence rate between the two types of social learning agents. AG\_SOCIAL\_PE2 learn faster than AG\_SOCIAL\_PE3 agents for 8, 12 and 16 attributes. However at 20 attributes the cultural learning agents outperform those learning via observation from a distance. The trend continues at 24 attributes as AG\_SOCIAL\_PE3 agents learn even faster.

In our previous study it was demonstrated that learning socially outperforms individual learning therefore, it is no surprise AG\_SOCIAL\_PE2 agents do better than AG\_GA\_PE1 agents. The fact that AG\_SOCIAL\_PE3 agents outperform AG\_GA\_PE1 agents supports the contention that artifact capability-learning via cultural evolution should proceed at a faster rate than through biological evolution. To understand the results that show agents learning via observation from a distance outperforming their cultural learning counterparts with simpler artifacts or artifacts with fewer attributes it must be remembered that these agents have partial knowledge of the artifact capability upfront. We believe that this partial knowledge gives AG\_SOCIAL\_PE2 agents a head start in the learning process. AG\_SOCIAL\_PE3 agents on the other hand begin with no knowledge of the capability and simply use the best of their social group to improve the process over time. According to Reynolds (1997)

knowledge compiled over time and maintained in the global belief space should guide the learning process such that it improves at every trial. As the number of attributes increase, the artifacts get more complex and the search space larger AG\_SOCIAL\_PE3 agents get better and eventually outperform AG\_SOCIAL\_PE2 agents. Although the observed threshold may vary and be problem dependent CA's have been used to optimize complex applications Chung and Reynolds (1998). Therefore we suggest that as an artifact gets more complex the likelihood that its capabilities would be best acquired via cultural learning increases especially when the visibility of attributes for observational learning is low.

## Conclusions and Future Work

In this study, we have designed and implemented a cultural evolutionary model supporting an agent with the objective of learning artifact capabilities without prior knowledge. Cultural learning agents benefit from belonging to a social network where individual experiences are shared. The model was designed by integrating a genetic and cultural algorithm with the framework of an artifact capability-learning agent. One of the objectives was to enhance the learning capacities of a previously implemented learning agent through cultural learning. Another objective was to compare cultural learning of artifact capabilities to observational learning from a distance. On a larger scale, we maintain that understanding the acquisition and evolution of artifact capabilities for single rational agents is a vital step towards representing their capacity to combine them into group capabilities, towards the accomplishment of more complex goals.

Results obtained from our multi-agent simulation implementation confirm that social learning outperforms individual learning and suggest that complex artifact capabilities are best learned via cultural learning. Although observational learning from a distance surpassed cultural learning for simpler artifacts, the fact that it requires access to an agent already in possession of the capability is a drawback. Additionally the agent must know how to copy the visible attributes with some degree of certainty. A cultural learning agent needs no capable agent in its vicinity and can begin the learning process without possessing any aspect of the artifact capability.

We believe that further experiments are necessary to investigate varying degrees of attribute visibility for agents learning via observation compared to the cultural learning process. One of the knowledge sources identified for the global belief space by (Reynolds, 1979) that was not used in this study is domain knowledge. For future work it can be used to influence the choice of goals for agents to pursue with regard to learning an artifact capability. It would be useful to simulate how goals evolve. One fitness function would no longer be sufficient in the learning process. The choice of a fitness function would be driven by the arti-

fact capability being learned, even for the same artifact. As an example, a knife can be used both as a cooking utensil and as a weapon. An agent's choice of one versus the other would require a different fitness function.

## Acknowledgements

This work is made possible by a grant from the National Science Foundation and NSERC Discovery No. 327482.

## References

- Acay, D. L., Tildar, G., and Sonenberg, L. (2008). Extending agent capabilities: Tools vs Agents. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 2, pages 259–265. IEEE Computer Society, Los Alamitos, CA.
- Acerbi, A. and Nolfi, S. (2007). Social learning and cultural evolution in embodied and situated agents. In *Proceedings of the First IEEE Symposium on Artificial Life*, pages 333–340. IEEE Press, Piscataway, NJ.
- Bacher, K., Allen, S., Lindholm, A. K., Bejder, L., and Krützen, M. (2010). Genes or culture: Are mitochondrial genes associated with tool use in bottlenose dolphins (*Tursiops* sp.)? *Behavior Genetics*, 40:706–714.
- Bluethmann, W., Ambrose, R., Diftler, M., Askew, S., Huber, E., Goza, M., Rehnmark, F., Lovchik, C., and Magruder, D. (2003). Robonaut: A robot designed to work with humans in space. *Autonomous Robots*, 14(2):179–197.
- Bratman, M. E. (1987). *Intentions, Plans and Practical Reason*. Harvard University Press, Cambridge, MA.
- Chung, C.-J. and Reynolds, R. G. (1998). CAEP: an evolution-based tool for real-valued function optimization using cultural algorithms. *International Journal on Artificial Intelligence Tools*, 7(3):239–291.
- Curran, D. and O’Riordan, C. (2007). The effects of cultural learning in populations of neural networks. *Artificial Life*, 13(1):45–67.
- Mokom, F. and Kobti, Z. (2011). Evolution of artifact capabilities. To appear in *Proceedings of the IEEE Congress on Evolutionary Computation*.
- Noble, J. and Franks, D. W. (2002). Social learning mechanisms compared in a simple environment. In *Artificial Life VIII*, pages 379–385. MIT Press, Cambridge, MA.
- Petroski, H. (1992). *The evolution of useful things*. Knopf, New York, NY.
- Plummer, T. (2004). Flaked stones and old bones: biological and cultural evolution at the dawn of technology. *American Journal of Physical Anthropology*, 125:118–164.
- Power, T. G. (2000). *Play and Exploration in Children and Animals*. Lawrence Erlbaum Associates, Mahwah, NJ.
- Preston, B. (1998). Cognition and tool use. *Mind and Language*, 13(4):513–547.
- Reynolds, R. G. (1979). *An Adaptive Computer Model of the Evolution of Agriculture for Hunter-gatherers in the Valley of Oaxaca*. PhD thesis, Dept. of Computer Science, University of Michigan.
- Reynolds, R. G. (1997). *Time, Process, and Structured Transformation in Archaeology*, chapter Why Does Cultural Evolution Proceed at a Faster Rate Than Biological Evolution, pages 269–282. Routledge Press, New York, NY.
- Reynolds, R. G. and Peng, B. (2004). Cultural algorithms: Modeling of how cultures learn to solve problems. In *Proceedings of the Sixteenth IEEE International Conference on Tools with Artificial Intelligence*, pages 166–172. IEEE Computer Society, Washington, DC.
- Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ.
- Schäfer, B. and Bergfeldt, N. (2007). Evolution of tool use behavior. In *Proceedings of the First IEEE Symposium on Artificial Life*, pages 31–38. IEEE Press, Piscataway, NJ.
- Stoytchev, A. (2005). Behavior-grounded representation of tool affordances. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3060–3065.
- Tomasello, M., Kruger, A. C., and Ratner, H. H. (1993). Cultural learning. *Behavioral and Brain Sciences*, 16:495–552.
- Wood, A. B., Horton, T. E., and Amant, R. S. (2005). Effective tool use in a habile agent. In Bass, E. J., editor, *Proceedings of the IEEE Systems and Information Engineering Design Symposium*, pages 75–81. IEEE.
- Wooldridge, M. J. (2000). *Reasoning about rational agents*. Intelligent Robots and Autonomous Agents. The MIT Press, Cambridge, MA.



# Surviving the Tragedy of Commons : Emergence of Altruism in a Population of Evolving Autonomous Agents

Jean-Marc Montanier<sup>1</sup> and Nicolas Bredeche<sup>1</sup>

<sup>1</sup>TAO - Univ. Paris-Sud, INRIA, CNRS, F-91405 Orsay, France

## Abstract

This paper explores the following question: how a fixed-size population of autonomous agents (such as a swarm of robotic agents) may evolve altruistic behaviors during open-ended evolution. In particular, we focus on a situation where the tragedy of commons can possibly occur: a situation where individuals must display altruistic behaviors in order for the whole population to avoid extinction. Our approach considers a sub-individual framework, defined at the level of genomes rather than agents, in order to provide an efficient algorithmic solution for the emergent of coordination among the population. Experiments show that the proposed evolutionary adaptation algorithm favors the emergence of altruistic behavior under some assumptions regarding genome relatedness. In-depth experimental studies explore the relation between genotypic diversity and degree of altruism as well as the exact nature of the evolutionary adaptation process.

## Introduction

Altruism is a remarkable behavior observed in Nature, where actions of an individual benefit other individuals even though these actions may negatively impact the individual's chances of survival. A well-known example is given by individuals that watch out for a predator and signal danger to the group whenever it is required, thus potentially drawing the predator's attention to them. The reason why some individuals may sacrifice themselves for the benefit of the group has long been studied and there are now some widely accepted theoretical basis regarding the relation between genotypic relatedness among individuals and degree of altruism, as first described by Hamilton (1964). Altruism has long been actively studied from Biology to Economics, from Sociology to Game Theory, to cite a few domains. It differs from cooperation as altruism requires no direct benefit nor reciprocity. Moreover, its benefit can only be measured at the level of the population, as summarized by Lehmann and Keller (2006).

This paper is concerned with the emergence of altruism in a fixed-size population of evolving autonomous agents where the environment is such that selfish behaviors lead to extinction. This situation is known as the tragedy of (unmanaged) commons, as introduced by Hardin (1968, 1994):

individuals must share a common limited resource, and possibly sacrifice their own benefit, so that the population survives through generations.

The main motivation behind this research is to propose a practical implementation of evolutionary adaptation in *a priori* unknown environments in the scope of a fixed-size population of autonomous agents. This assumption is central to our motivation as the long term goal is to provide practical algorithmic solutions that can be deployed in a swarm of virtual agents in complex environments as well as real world autonomous robots. The contribution in this paper is then both fundamental and practical as the emergence of altruism during the course of evolution is experimentally studied, with a particular focus on its causes and consequences, and is considered within an experimental setup that is closely related to the target application: a 2D virtual environment with realistic assumptions inspired from autonomous robotics.

The paper is organized as follow: the definitions of altruism and tragedy of commons are provided in the next section, along with a short description of relevant contributions from the fields of Artificial Life and Evolutionary Robotics. Then, the environment-driven evolutionary adaptation algorithm is described as well as the experimental settings used for the experiment. Results from the experiment are given and discussed, with a particular focus on the nature of altruism observed. Finally, the last section provides a discussion and conclusion and sketches future directions for this work.

## Context and Motivation

This section starts with a definition of the Tragedy of Commons, a well-known social dilemma where the population welfare strongly depends on individual behaviors. Then, a definition of altruism is given as well as a brief overview of its theoretical foundations in Biology. The section ends with a short review of related works in the field of Artificial Life and Evolutionary Robotics.

## The Tragedy of Commons

The tragedy of (unmanaged) commons ((Hardin, 1968, 1994)) is a particular kind of social dilemma where a pop-

ulation of individuals have access to a finite common resource pool: each individual may temporarily increase its fitness through selfish behavior, but this inevitably leads to exhaust the common resource pool, ultimately ending with population extinction. The classic example describes farmers optimizing their personal benefit by owning as many cows as possible without any regards for the common grazing the cows feed from, which will quickly suffer from over-exploitation, ending with cows dying from starvation.

The tragedy of commons has been widely studied in both Evolutionary Biology and Economics (Mankiw, 2009). Using a terminology from Economics, the conditions for the occurrence of the tragedy of commons requires that the resource must be accessible to anyone ("non-excludable") but in limited quantity, thus implying competition ("rivalry") among individuals. It shares some similarities with the well-known public goods dilemma<sup>1</sup> regarding the condition of unrestricted accessibility to the resource, but also differs as the subtractability of the resource may penalize the survival rate of the population (e.g. because of free-riders). From the Biology viewpoint, the tragedy of commons is known to be responsible of in-group competition among individuals.

A possible explanation for the tragedy of commons is the negative impact of reciprocity, where free-riders are favored as they focus on their own personal fitness gain with no regards to the cost at the level of the population (Sober, 1992). However, several strategies have been identified and discussed in the literature for "solving" the tragedy of commons: kin selection, policing (self-regulated punishment) or diminishing returns (population behavior depends on ecological feedback) are all good candidates observed in Nature (Rankin et al. (2007)).

## Definition of Altruism

The emergence of cooperation and altruism has been the focus of a particular attention from many research fields, including of course Biology.

The distinction between cooperation with mutual benefit<sup>2</sup> (West et al., 2007) and "strong" altruism (termed altruism from now on) depends on the nature of the fitness benefit at the level of *either* the individual *or* the population (Lehmann and Keller, 2006). Cooperation implies that a given individual benefits from its behavior during its lifetime, either through direct or delayed (i.e. through repeated interactions) reciprocity. Altruism, on the other hand, characterizes the sacrifice of (part of) one own's fitness for the benefit of others. Therefore, an altruistic behavior bene-

fits other individuals and possibly has a positive impact on longer time-scale (e.g. more than a single lifetime).

Several theories have been identified, covering different kinds of behavior observed in Nature, from mutualism to conditional cooperation. On the one hand, mutualism is the case where cooperation leads to direct benefit even though a single individual displays a cooperative behavior (Maynard Smith J., 1983; Lima, 1989; Packer C., 1988; Dugatkin and Wilson, 1992). On the other hand, the more classic conditional cooperation scheme implies that all individuals share the same cooperative strategy so that the whole population welfare is increased: kin selection (Hamilton, 1964; Maynard Smith, 1964), reciprocity (Trivers, 1971; Axelrod and Hamilton, 1981) or the more controversial group selection (Wynne-Edwards, 1986; Dugatkin, 1994; West et al., 2007) can be accounted for such conditional cooperation.

While the emergence of cooperation can be explained by the fact that every individuals benefit from such a behavior (i.e. no cost to cooperate), the justification for altruism is not as straight-forward. The idea of inclusive fitness proposed by Hamilton (1964) is now widely accepted to account for the emergence of altruism: inclusive fitness considers the fitness of a particular individual to depend both on its own behavior and the behavior of its close relatives. The basic idea is to consider individuals as vehicles for genes, therefore kinship must be taken into account rather than the sole interest of one individual/vehicle. Of course, sacrificing oneself depends on several parameters such as the expected fitness loss (from sacrifice) and benefit (for others) as well as the genotypic relatedness of the individuals concerned (closer relatives may imply increased altruistic behaviors).

Hamilton formalized the relationship between cost, benefit and relatedness in the following equation:  $C/B < r$ . The Cost  $C$  is the amount of fitness lost by an altruistic individual. The benefit  $B$  is the amount of fitness gained by the recipient that benefits from the altruistic behavior. And  $r$  is the genotypic relatedness between the two individuals. The term *kin selection* has been introduced by Maynard Smith (1964) to illustrate the mechanism and consequences with inclusive fitness: if one's individual is willing to sacrifice itself for closely related individuals, the gene responsible for such an altruistic behavior may spread through natural selection as it is likely to be present also in the genotypic material of its parents.

## Models of Altruism in Artificial Life

Altruistic behavior, as well as the emergence of altruism, has also been investigated in the field of Artificial Life. All the major theories have been studied: kin selection (Sober, 1992; Leticia et al., 2004), group selection (Fletcher and Zwick, 2004, 2007) and other mechanisms such as effect of increased environment's viscosity (Mitteldorf and Wilson, 2000), communication (Ackley and Littman, 1994) and tag mechanism (Spector et al., 2004; Spector and Klein,

<sup>1</sup>In the public goods dilemma, individuals may choose to invest a part of their benefit for the group welfare.

<sup>2</sup>Cooperation is also sometimes used as a synonym for altruism (e.g. cooperation in the prisoner's dilemma corresponds to altruism (Sober, 1992)). In this paper, we assume the restricted and well-accepted definition of cooperation as a behavior leading to mutual benefit.

2006). Previous works have provided studies with various approaches, from game theoretic models to discrete and continuous virtual world simulations. Moreover, kin selection, reciprocity and group selection have been described as variations of a similar mechanism favoring the correlation of interaction between agents (Woodcock and Heath, 2002).

The emergence of altruism under specific condition have also been studied in virtual or real environments, in particular with respect to the public good dilemma (Connelly et al., 2010; Waibel et al., 2009) and to the tragedy of commons (Spector et al., 2004; Scogings and Hawick, 2008), with similar concerns for different selection schemes.

Waibel et al. (2009) discusses the ability to evolve altruism in team of homogeneous robots with group selection in a setup similar to the public good dilemma. Facing the same environmental conditions, Connelly et al. (2010) experimentally show that altruism naturally emerges as long as resources is widely available.

The tragedy of commons has been addressed by Spector et al. (2004), where tag recognition favors the interaction between altruistic agents facing a tragedy of commons, and by Scogings and Hawick (2008) in a prey-predator setup. Even though their work considered population with fixed strategy (rather than evolutionary adaptation), they illustrated the ability of altruistic population to survive in aggressive environment even when confronted to selfish individuals.

## Method

In this paper, we are interested in identifying the emergence of altruism in the scope of environment-driven self-adaptation in a population of autonomous agents. The motivation behind this work is two-fold. Firstly, our long-term motivation targets the design of an evolutionary adaptation algorithm for a limited group of autonomous agents that is capable of facing *a priori* unknown situations. An important requirement is that the algorithm should be implementable in a virtual or real-world environment (e.g. multi-agent simulation, agents in virtual worlds, robot swarms).

Secondly, we ask the following question: what can be expected when a population of evolving agents faces the tragedy of commons. This implies to identify if a strategy emerges, but also the nature of this strategy, if any.

In this section, we describe the algorithm and the experimental setting used for this work. In particular, the experimental setting has been designed so that the population faces a setup where the tragedy of commons is expected to occur. Lastly, methodological tools for monitoring altruistic behaviors are introduced at the end of the section.

## Algorithm

The mEDEA<sup>3</sup> algorithm takes inspiration from the selfish gene metaphor popularized by Dawkins (1976) and performs as an evolutionary adaptation algorithm that can be

<sup>3</sup>*minimal Environment-driven Distributed Evol. Adaptation.*

distributed over a population of robotic agents (i.e. each agent in the population runs the same algorithm, but carries different genomes). It was first introduced by Bredeche and Montanier (2010) to address robustness issue with dynamic unknown environments and has been successfully validated on real e-puck autonomous robots (Bredeche et al. (2011)).

In this framework, each agent contains an *active* genome, which (indirectly) controls the agent's behavior, and a *reservoir of stored genomes*, which is empty at first. At each time step, each agent *broadcasts* in a limited range (approx.  $1/32^{th}$  of the arena's width) a *slightly* mutated copy of its active genome (e.g. with gaussian mutation) and stores genomes received from neighbors, if close enough. At the end of a "lifetime" (i.e. a pre-defined number of time steps), each agent "forgets" its active genome and *randomly* picks one genome from its reservoir of stored genomes (if not empty). Then the reservoir is emptied, and a new lifetime starts. This algorithm is duplicated within each agent in the population, even though agents' behaviors differ depending on each agent's current active genome.

There are three major claims why this algorithm works. Firstly, selection pressure occurs at the population level (the more a genome spreads itself, the higher its fitness) rather than at the individual level (random sampling). Secondly, genomes survive only through spreading (as an active genome is automatically deleted locally at the end of a generation). Thirdly, individual fitness improves over time as conservative variations generate new candidates that explore alternative (but closely related) behavioral strategies.

In practical, this algorithm provides an evolutionary adaptation mechanism, but does not provide a control function. The actual control of the agent behavior shall be performed by a dedicated controller whose parameters are determined from the genome. In other words, the mEDEA algorithm provides evolutionary adaptation by tuning the control architecture. In the rest of this paper, the controller used is a Multilayer Perceptron whose weights are decoded from the genome (more details in the next Section).

The mEDEA algorithm shares some similarity with the basic concepts demonstrated in Tierra (Ray, 1991), AVIDA (Adami et al., 1994) and followers, but also differs as it was originally designed for real world environments with a limited number of moving autonomous agents such as mobile robots. It can also be related to Embodied Evolutionary Robotics (Watson et al., 2002) regarding the possible implementation on physical agents, but with the major difference that it is not meant to optimize a pre-defined objective function.

## Experimental Setup

In order to account for the existence of altruism, we have defined a foraging task where a population of autonomous agents must eat food items to maintain a positive energy level. The experimental setup used in the next section is il-

illustrated in figure 1, with food items (circles), agents (small dots) and obstacles. The environment and task depends on the following elements: (1) *Self-sustainability*: foraging is necessary to survive, as each food items give a small amount of battery energy. However, an agent's battery is limited to a maximum amount of energy, and foraging may end up in wasting resource. (2) *Foraging behavior*: an agent may choose to harvest *all* or *part* of a food item. (3) *Re-grow rate*: whenever a food item is harvested, it is removed from the environment until it grows back after some delay. The time to grow back depends on the quantity of energy harvested from the food item.

As a consequence, the environment features a common resource pool for which agents compete: a perfect setup for the Tragedy of Commons to occur. Indeed, it is then enough to set the appropriate delay before a given food item would grow back. This is achieved by setting the maximum re-grow delay for a food item ( $EP_{LagMax}$ , with  $EP$  as in "Energy Point"), which in turn will be used to compute on-the-fly the re-grow delay of a food item that was just harvested ( $EP_{Lag}$ ). This is described in equation 1, which also takes into account the amount of energy harvested by an agent from the food item ( $E_{harvested}$ ) and the amount of energy available in each food item ( $EP_{eMax}$ ).

$$EP_{Lag} = E_{harvested} / EP_{eMax} * EP_{LagMax} \quad (1)$$

Within this setup, it is expected that altruistic agents in aggressive environments shall harvest the minimum amount of energy from each food items, therefore increasing the availability of the resource (short re-grow delay, no wasted energy). On the other hand, selfish behaviors are likely to be fitted for small values of  $EP_{LagMax}$ , but are expected to become more and more critical as the value of  $EP_{LagMax}$  increases.

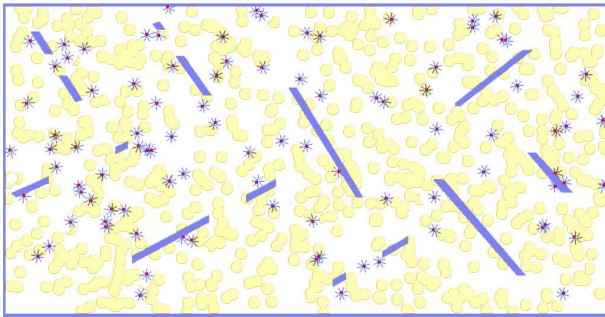


Figure 1: Snapshot from the simulator: food items (circles), agents (dots) and obstacles

## Methodology

In order to account for altruism, we define a measure for monitoring the *cost of altruism* for one foraging agent. In the setup described earlier, this corresponds to measuring the

amount of energy that *could be* consumed when harvesting a food item, but which is actually *not consumed* by the agent. This is formally defined in equation 2.

$$Cost = \max(0, \min(EP_{eMax}, r_{E_{max}} - r_{E_{now}}) - E_{harvested}) \quad (2)$$

Where  $EP_{eMax}$  is defined as before (i.e. maximal energy in a food item),  $r_{E_{max}}$  is the maximal energy level of an agent,  $r_{E_{now}}$  is the current energy level of the agent and  $E_{harvested}$  is the energy harvested by the agent from the food item.

While a selfish agent shall have a cost of zero, an altruistic agent should be able to perform a trade-off between its altruistic nature and its survival needs. Therefore, the cost of altruism can be seen as the agent's level of sacrifice which is continuous (a quantity of energy) rather than discrete (eat or don't eat).

## Results and Analysis

This section presents results obtained running the mEDEA algorithm in the environment described in the previous section. The organization of the section is as follow: the algorithm is evaluated for its ability to evolve agents with altruistic behavior. Then, the nature of altruistic behavior is investigated, considering the balance between environmental pressure and the algorithm's mechanisms. Finally, the relation between genotypic relatedness and the degree of altruism is explored along with its impact on the survival rate of the population.

All experiments were conducted with 100 robotic agents in the environment described and illustrated in the previous section. The environment contains 800 food items and an agent may harvest a maximum of 50 units from a food item. Each agent consumes 1 unit of energy per step, and can store up to 800 energy units (harvesting surplus is lost). If the agent's battery level drops to zero, the agent stops and its genome is lost. It is then refilled with a small portion of energy, but remains still until it receives a new genome.

The control architecture is a Multilayer Perceptron (MLP) with 5 hidden neurons, 11 inputs (8 proximity sensors, battery level and orientation/distance to the closest food item) and 3 outputs (left/right motor and proportion of energy to be harvested from a food item, if any). The weights of the MLP are decoded from the active genome of the agent. Each agent broadcasts a mutated copy of its own genome and receives genomes from neighbors within a limited range (roughly 1/10th of the length of the larger side of the environment). The mutation operator used in the Medea algorithm is defined as a gaussian mutation with a  $\sigma$  parameter.  $\sigma$  is included into the genome (i.e. similar to a self-adaptive Evolution Strategy) and ranges from 0.01 (low mutation rate) to 0.5 (large mutation rate).

All results shown here have been achieved in ROBORO,

a fast 2D simulation for robotic agents, originally introduced by Bredeche and Montanier (2010). The source code for reproducing the experiments is freely available for download (<http://www.lri.fr/~montanier/roborobo-ecal>). For each experimental settings, a set of 600 independent runs have been performed during 320000 iterations (= 800 generations) to provide statistically significant data.

## Emergence of Altruism in Medea

A large set of experiments was performed under various environmental pressures by setting a specific value of  $EP_{LagMax}$  for each run, ranging from 25 steps (easy environment) to 400 steps (aggressive environment), for a total of 16 setups. For each setup (i.e. a fixed value of  $EP_{LagMax}$ ), 600 independent runs were performed and results were aggregated to extract various indicators: number of active agents, average cost measure and energy balance (i.e. a positive value means agents harvest more than the minimal requirement). In all experiments, the course of evolution is similar: the number of active agents quickly increases to a stable value while costs start from random values and stabilize to (possibly) positive values. While the increasing number of active agents is expected from evolutionary adaptation, the second observation is of primary importance regarding the possibility of altruistic behavior: a positive cost value would imply that agents do not systematically harvest all possible energy from the food items.

Results are summarized in figures 3(a), 3(b) and 2 (resp. number of active agents, cost measure and energy balance), by taking into consideration the last 10 generations of all runs for each setup (i.e. after convergence to stable behaviors). Altruistic behavior in the context of increasing environmental pressure can be observed by looking at the cost, which converges to a stable value, while the energy balance converges to zero (i.e. the limit for survival). Indeed, altruistic behaviors are observed starting with environments with  $EP_{LagMax} = 100$ , and remains afterwards. With stronger environmental pressures (larger values of  $EP_{LagMax}$ ), the number of active agents decreases, which confirms that the environment is becoming more and more challenging.

Several observations can be drawn from these results. Firstly, altruistic behaviors are difficult to observe when environmental pressure is low and tragedy of commons not bound to occur (median values are close to zero for values of  $EP_{LagMax}$  under 100 steps). This tends to reveal the greedy nature of the algorithm: without environmental pressure, altruism does not emerge spontaneously. In fact, it is possible to classify the individuals' behavioral patterns with respect to (a) their fellow agents (*selfish* vs. *altruistic* behavior) and (b) the environment (*frugal* vs. *greedy* behavior). the mEDEA algorithm tends to generate *greedy* but *altruistic* agents depending on the environment at hand. Secondly, altruistic behaviors remain stable in the population even though the environmental pressure increases and

the number of active agents starts to drop, implying limited correlation between the level of altruism and environmental pressure. This is explored in the following.

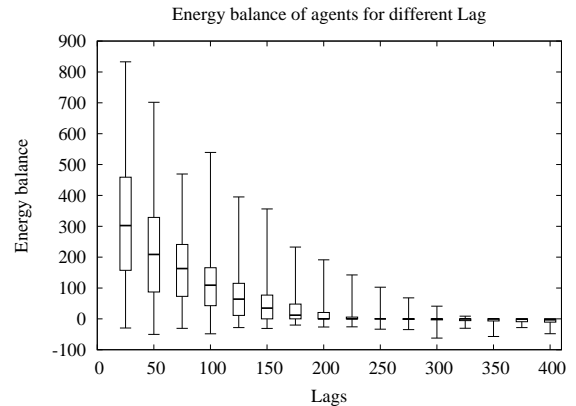


Figure 2: Results with  $EP_{LagMax}$  between 25 and 400 : Energy Balance (data: boxplots are drawn from the median values from each run, i.e. for each run, some agents (not shown) are likely to have larger positive energy balances)

## Investigating the Nature of Altruism

In order to explore the dynamics of the algorithm, a first experiment is designed to evaluate its ability to converge towards the same results from different initial conditions. Starting with a population of agents already evolved in a challenging setup ( $EP_{LagMax} = 400$ , strong pressure, used during 1000 generations), the population is abruptly changed to a smoother environment ( $EP_{LagMax} = 200$ , moderate pressure) and re-adaptation (if any) is studied. The expected outcome is that the number of active agents and the cost measure should converge back to the expected values (shown before). This is indeed what is observed, as shown in figure 4, advocating for the robustness with regards to initial conditions, at least in this case (i.e. starting from already evolved genomes rather than pure random genomes). This is also confirmed by a Mann-Whitney's statistical test.

However, a careful analysis of the results reveals a surprising feature occurring when the environmental pressure is changed: the number of active agents rises significantly before going back down to its final stable value. The same holds for the cost measure, as a sudden drop is observed, preceding a slow convergence to the expected, higher, value. This is indeed a surprise as, for a brief moment, individuals actually have a better survival rate even though more egoistic behaviors are monitored. A closer look at the results in the close vicinity of the change in the environment (not visible at this resolution) actually confirms this: after the environmental change, the number of active agents (resp. cost measure) quickly rises (resp. drops), before slowly converging back

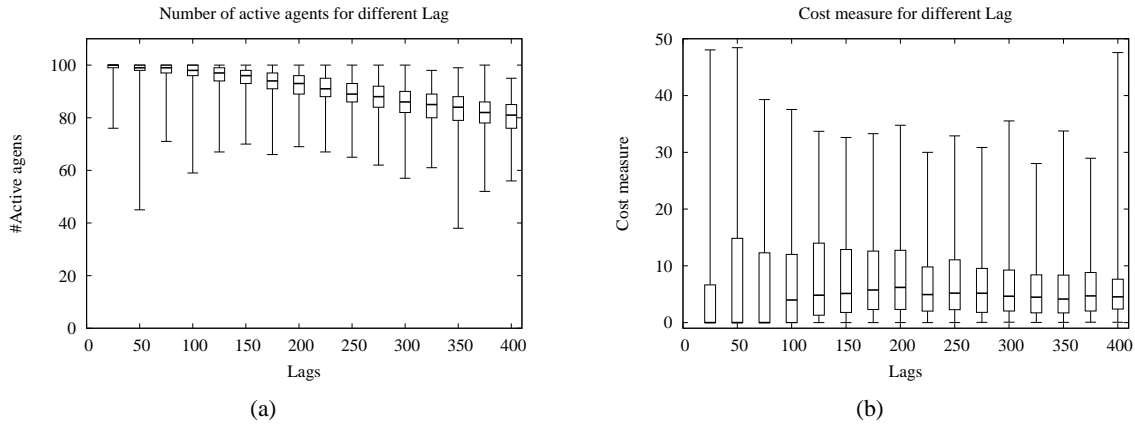


Figure 3: Results with  $EP_{LagMax}$  between 25 and 400: *a*) Number of active robots (data: value from each run) ; *b*) Cost measure (data: median values from each run)

to its final expected value.

A candidate hypothesis for explaining the algorithm's behavior is to reconsider the very nature of *what* can be stated as its intrinsic motivation: mEDEA may be performing a trade-off between survival *and* stability of evolutionary dynamics, rather than survival only. In order to investigate this hypothesis, we define a measure of evolutionary stability that takes into account the number of ancestors from a previous generation for individuals of the current generation (i.e. the larger the number, the more the ancestor with one offspring only). Larger numbers imply a more stable population as it means that more genomes actually survived through their offsprings. In other words, a population with many ancestors imply lack of selective pressure. In practical, this is defined as follow:  $nbStrains_{gen=N-b}/nbActiveAgents_{gen=N}$ , with  $nbStrains$  the number of ancestors from  $b$  generations ago with at least one descendant in the current generation. The value is normalized in  $[0, 1]$ . Lower values imply increased selective pressure.

Figure 5 tracks this value for a few generations: for each generation (i.e. each boxplot), the (normalized) number of ancestors from  $b = 10$  generations ago with at least one offspring in the current generation is drawn. During the short increase in performance after the environmental change, the number of ancestors decreases for at least 10 generations, which indicates that fewer genomes actually benefited from a stronger selective advantage. However, selective pressure then goes back to a more conservative level, even though behaviors end up being sub-optimal with respect to survival (as shown before). Why the best genomes for survival do not remain in the population is yet to be fully understood. In this context, it is likely that egoistic agents may only temporarily benefit from the change, as they may not be enough in numbers to take over the population before altruistic agents adapt to the new environment. Indeed, very specific initial con-

ditions (forcing egoistic behavior at start-up) or dedicated mechanisms in the algorithm (see next section for a discussion) may be required to obtain the best population wrt. surviving rate.

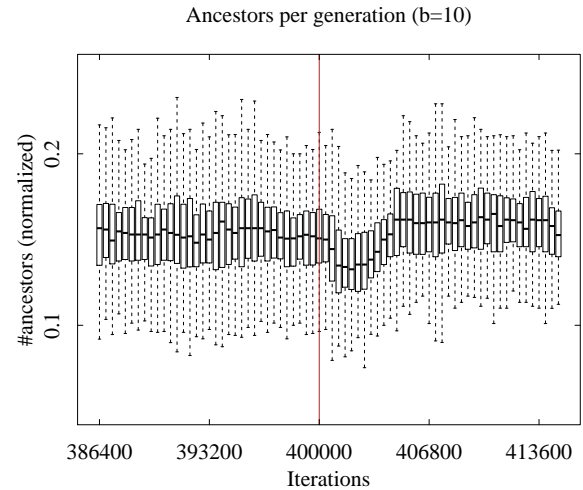


Figure 5: Ancestors from generation  $N - 10$  with at least one offspring in the current generation (34 generations before and after the change are shown).

## Discussion on Diversity and Altruism

As stated previously, it is likely that selective pressure acts in favor of a trade-off between optimizing survival and algorithmic internal stability. But what happens if one were to deliberately enforce genotypic homogeneity? In the following, we address this question and discuss its possible implications. The motivation is two-fold: firstly, the goal is to explore the relation between genotypic homogeneity, level of altruism and survival rate. Secondly, part of the answer



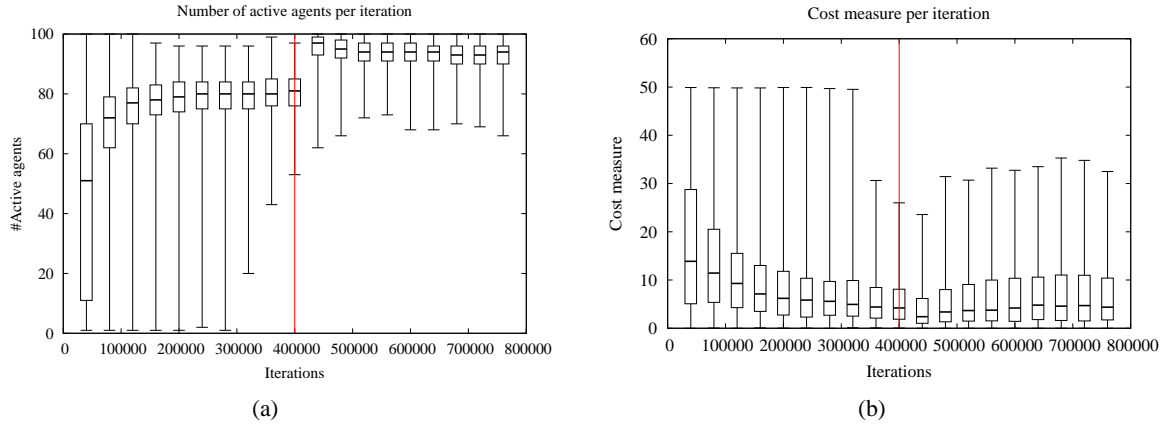


Figure 4: Environment change from strong to moderate pressure (see text). *a)* Number of active robots ; *b)* Cost measure

to this question is a first step towards controlling the evolutionary dynamics at work in the algorithm.

A set of additional experiments have been performed where genotypic relatedness is favored during the selection process, in order to decrease genotypic distance among individuals in the population. In practical, the algorithm's random selection that is embedded in each agent is replaced by a tournament selection (Miller and Goldberg, 1995) (also embedded in each agent), where ranking is based on the genotypic (euclidian) distance between the previously active genome and the locally available genomes (the closer, the better). Tournament selection combined with genotypic distance (termed kin-tournament from now on) makes it possible to introduce an explicit pressure towards kin selection, which can easily be tuned by the size of the tournament.

Experiments with a tournament size of 3 (roughly corresponding to medium pressure towards kin selection) have been achieved with two setups, one with moderate environmental pressure ( $EP_{LagMax} = 200$ ) and the other with a strong pressure ( $EP_{LagMax} = 400$ ). For each setup, 200 runs were performed, and statistical test are computed with Mann-Whitney's Test to clearly establish the difference in performance. Performing kin selection increases the level of altruism in both cases (roughly doubling it,  $p - value < 10^{-15}$ ). While the number of runs with extinctions is roughly similar ( $p - value = 0.07$  for  $EP_{LagMax} = 200$ , and  $p - value = 0.71$  for  $EP_{LagMax} = 400$ ), enforced kin selection suffers from a smaller number of active agents ( $p - value < 10^{-15}$ ).

These results can be put in perspective with Hamilton's idea of inclusive fitness (Hamilton, 1964). The intrinsic mechanisms in the algorithm, in particular conservative mutation, already imply a strong genotypic relation between one genome and its offsprings. Kin selection is shown to artificially increase the already existing level of altruism, at the cost of a decreased overall performance wrt. to individual survival. This is not a surprise as altruistic behaviors

were already shown previously to lead to sub-optimal survival rate, which is even more critical when environmental pressure is aggressive. Nevertheless, the kin-tournament selection as proposed here actually does provide an interesting tunable mechanism to act on the level of altruism, and could possibly lead to a more competitive, heterogeneous population if kin selection is penalized rather than favored.

## Conclusions and Perspectives

In this paper, we investigated evolutionary adaptation in a population of robotic agents whenever altruistic behaviors are mandatory to survive. The algorithm under scrutiny was shown to naturally evolve greedy-altruistic agents within aggressive environments (ie. greedy behavior whenever it does not impact the survival rate of the population). An important message from this paper is that evolutionary adaptation in this context does not automatically lead to the best survival strategy but rather converge towards a trade-off between algorithmic stability and survival. Also, the relation between genotypic relatedness and the level of altruism was confirmed and a possible mechanism to control the level of altruism has been identified.

Perspectives from this work include deeper investigation regarding the exact causes of the sub-optimal survival strategies obtained. Moreover, tuning the level of altruism offers interesting perspectives with regards to modeling environmental-feedback induced altruistic behaviors, such as *diminishing returns*, where altruism may be regulated by the environment (Rankin et al., 2007).

## Acknowledgements

This work was made possible by the European Union FET Proactive Initiative: Pervasive Adaptation funding the Symbion project under grant agreement 216342. Some of the experiments presented in this paper were carried out using the Grid'5000 experimental testbed (cf. <https://www.grid5000.fr>). We'd like to thank Simon Carrignon for his last minute (much appreciated) help with generating figure 5.



## References

- Ackley, D. H. and Littman, M. L. (1994). Altruism in the evolution of communication. In Brooks, R. and Maes, P., editors, *Artificial Life IV*, pages 40–48, Cambridge, MA. MIT Press.
- Adami, C., Brown, C. T., and Kellogg, W. K. (1994). Evolutionary Learning in the 2D Artificial Life System Avida. In *Artificial Life IV*, pages 377–381.
- Axelrod, R. and Hamilton, W. D. (1981). The evolution of cooperation. *Science*, 211(4489):1390–1396.
- Bredeche, N. and Montanier, J.-M. (2010). Environment-driven Embodied Evolution in a Population of Autonomous Agents. In Springer, editor, *The 11th International Conference on Parallel Problem Solving From Nature (PPSN 2010)*, pages 290–299.
- Bredeche, N., Montanier, J.-M., Wenguo, L., and Alan FT, W. (2011). Environment-driven Distributed Evolutionary Adaptation in a Population of Autonomous Robotic Agents. *Mathematical and Computer Modelling of Dynamical Systems*.
- Connelly, B. D., Beckmann, B. E., and McKinley, P. K. (2010). Resource abundance promotes the evolution of public goods cooperation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 143–150.
- Dawkins, R. (1976). *The Selfish Gene*. Oxford University Press.
- Dugatkin, L. A. (1994). Subtle ways of shifting the balance in favor of between-group selection. *Behavioral and Brain Sciences*, 17:618–619.
- Dugatkin, L. A. and Wilson, D. S. (1992). The prerequisites for strategic behaviour in bluegill sunfish, *lepomis macrochirus*. *Animal Behaviour*, 44(Part 2):223 – 230.
- Fletcher, J. A. and Zwick, M. (2004). Strong altruism can evolve in randomly formed groups. *Journal of Theoretical Biology*, 228(3):303 – 313.
- Fletcher, J. A. and Zwick, M. (2007). The evolution of altruism: Game theory in multilevel selection and inclusive fitness. *Journal of Theoretical Biology*, 245(1):26 – 36.
- Hamilton, W. (1964). The genetical evolution of social behaviour. i. *Journal of Theoretical Biology*, 7(1):1–16.
- Hardin, G. (1968). The tragedy of the commons. *Science*, 162:1243–1248.
- Hardin, G. (1994). The tragedy of the unmanaged commons. *Trends in Ecology & Evolution*, 9(5):199–199.
- Lehmann, L. and Keller, L. (2006). The evolution of cooperation and altruism - a general framework and a classification of models. *Journal of Evolutionary Biology*, 19(5):1365–1376.
- Leticia, A., A, F. J., and Cutter, A. D. (2004). The kin composition of social groups: trading group size for degree of altruism. *Am Nat*, 164(2):132–44.
- Lima, S. L. (1989). Iterated Prisoner’s Dilemma: An Approach to Evolutionarily Stable Cooperation. *American Naturalist*, 134:828–834.
- Mankiw (2009). *Principles of economics, 3rd edition*. South-Western College Pub, february edition.
- Maynard Smith, J. (1964). Kin selection and group selection. *Nature*, 201:1145.
- Maynard Smith J. (1983). *Game theory and the evolution of co-operation*, page 445 456. Cambridge University Press, Cambridge.
- Miller, B. L. and Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9:193–212.
- Mitteldorf, J. and Wilson, D. S. (2000). Population viscosity and the evolution of altruism. *Journal of Theoretical Biology*, pages 481–496.
- Packer C. (1988). Constraints on the evolution of reciprocity : lessons from cooperative hunting. *Ethology Sociobiology*, 9:137–147.
- Rankin, D., Bargum, K., and Kokko, H. (2007). The tragedy of the commons in evolutionary biology. *Trends in Ecology & Evolution*, 22(12):643–651.
- Ray, T. S. (1991). An approach to the synthesis of life. In Langton, C., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II*, volume XI of *Santa Fe Institute. Studies in the Sciences of Complexity*, pages 371–408. Addison-Wesley, Redwood City, CA.
- Scogings, C. and Hawick, K. (2008). Altruism amongst spatial predator-prey animats. In Bullock, S., Noble, J., Watson, R., and Bedau, M. A., editors, *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 537–544. MIT Press, Cambridge, MA.
- Sober, E. (1992). The evolution of altruism: Correlation, cost, and benefit. *Biology and Philosophy*, 7(2).
- Spector, L. and Klein, J. (2006). Genetic stability and territorial structure facilitate the evolution of tag-mediated altruism. *Artificial Life*, 12.
- Spector, L., Klein, J., and Perry, C. (2004). Tags and the evolution of cooperation in complex environments. In *Proceedings of the AAAI 2004 Symposium on Artificial Multiagent Learning*. AAAI Press.
- Trivers, R. L. (1971). The evolution of reciprocal altruism. *The Quarterly Review of Biology*, 46(1):35.
- Waibel, M., Keller, L., Floreano, D., and Member, S. (2009). Genetic team composition and level of selection in the evolution of multi-agent systems. *IEEE Transactions on Evolutionary Computation*.
- Watson, R., Ficici, S., and Pollack, J. (2002). Embodied evolution: Distributing an evolutionary algorithm in a population of robots.
- West, S. A., Griffin, A. S., and Gardner, A. (2007). Social semantics: altruism, cooperation, mutualism, strong reciprocity and group selection. *Journal of Evolutionary Biology*, 20:415–432.
- Woodcock, S. and Heath, J. (2002). The robustness of altruism as an evolutionary strategy. *Biology and Philosophy*, 17:567–590. 10.1023/A:1020598804674.
- Wynne-Edwards, V. C. (1986). *Evolution through group selection*. Blackwell Scientific, Oxford.

# Learning Symbolic Forward Models for Robotic Motion Planning and Control

Hiroataka Moriguchi<sup>1,2</sup> and Hod Lipson<sup>2</sup>

<sup>1</sup>Department of Computer Science, the University of Tokyo, Tokyo, Japan

<sup>2</sup>Creative Machines Lab, Cornell University, Ithaca, NY, USA  
hmori@nii.ac.jp, hod.lipson@cornell.edu

## Abstract

Physiological studies suggest that humans have internal dynamics models for both themselves as well as their environment, which are integral components in motion planning and control. Although robotic systems rely on similar models, a primary constraint for robotic applications is how such models are acquired and developed. Traditionally human engineers derive the dynamics models for robots; this approach is not scalable for increasingly complex designs. As a result, there is growing interest in model inference methods, which automate the modeling process and extends the design range of robots. This paper proposes a novel method that infers dynamics models as mathematical expressions via Symbolic Regression and applies them for robotic motion planning and control tasks. The advantage of this expression is not only the accuracy but also the computational efficiency. Experimental results on underpowered pendulum domains validate that our inferred models enable fast motion planning and real-time control based on rapid re-planning, with significantly superior results over Support Vector Regression and Gaussian Process Regression.

## Introduction

Recent advancement in robotics has resulted in multitude of morphologically diverse robots, ranging from joint-based, legged robots to soft, continuous robots. The traditional approach to designing robot controllers requires that human engineers derive a dynamics model using first principles and prior knowledge about robots. However, as these robots increase in complexity, obtaining the dynamics model using analytical methods becomes significantly more difficult. Instead, inferring a dynamics model via machine learning approaches is a promising alternative.

Physiological evidences suggest that humans also acquire dynamics models that are vital for motion planning and control (Wolpert et al., 1995). Such dynamics models are classified into two types: *forward* and *inverse models* (Kawato, 1999). Forward models predict the consequence of motor commands, while inverse models determine the necessary motor commands to achieve a desired state transition.

The goal of this work is to infer a robot's forward model and to effectively apply it in motion planning and control

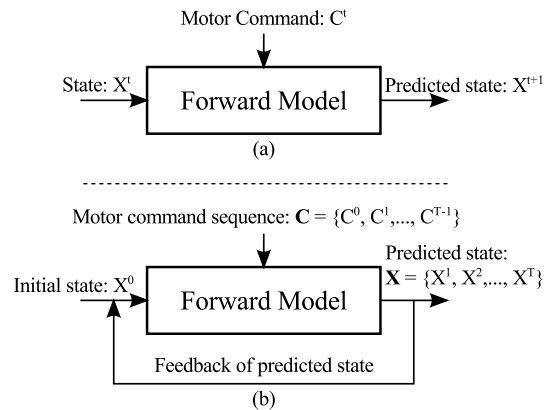


Figure 1: Diagrams of forward modeling. (a) One-step state prediction with forward model, and (b) iterative state prediction with internal feedback for simulating command sequences.

tasks. In robotics, forward models take current state and motor command as input, and predict the next state without actually executing the command (Fig. 1(a)). Furthermore, they can also simulate command sequences of arbitrary length by iterating one-step predictions with internal feedback loop (Fig. 1(b)). The latter provides an infrastructure for subsequent command optimization, which generally takes the form of motion or trajectory planning.

For applications in robotic motion planning, forward models should be accurate as well as computationally efficient. Model accuracy is essential as faulty prediction may lead to misleading optimization. Since simulating command sequences requires iterative use of forward models, even minor errors in individual predictions can accumulate, resulting in significant discrepancies over the course of the simulation. Real-time re-planning is an effective remedy to this problem; however this requires that the model is computationally lightweight for rapid evaluations under a strict time constraint.

This paper introduces a novel method to infer the forward model of an arbitrary robot and apply them for mo-

Table 1: Comparison of autonomous modeling methods for robotic motion planning and control.

Authors	Type of models	Algorithm	Usage
Sturm et al. (2008)	forward and inverse	GPR	feedback motion control
Nguyen-Tuong and Peters (2008)	forward and inverse	GPR	feedback motion control
Dearden and Demiris (2005)	forward	GPR	state prediction
Bongard et al. (2006)	morphological	EA	motion planning (offline)
Ours	forward	SR	motion planning (offline, real-time)

tion planning and control problems. Our method uses Symbolic Regression (SR) (Koza, 1992) for model inference. Models inferred via SR are mathematical expressions that accurately explain robot’s dynamics and are computationally lightweight. Experiments on underpowered pendulums show that the accuracy of our models are comparable to those learned with Gaussian Process Regression (GPR), while being superior to those with Support Vector Regression (SVR). Another advantage of SR models is that they can be evaluated for prediction at least three orders of magnitude faster than GPR and SVR models. This allows for fast motion planning and real-time control based on rapid re-planning. For motion planning, our method can find a more desirable plan with smaller computational effort compared to GPR and SVR-based methods. Furthermore, our method can achieve large performance gain via real-time re-planning, while methods based on GPR or SVR models cannot meet strict time constraints.

## Background and Related Work

There are a wide range of approaches and applications for autonomous modeling of robots’ dynamics. This section provides a brief survey of previous studies, summarized in Table 1.

Sturm et al. (2008) investigated an autonomous modeling approach that used Gaussian processes to infer the dynamics model of robot arms. Their models inferred the relationship between the motor targets of all joints and resulting pose of the arm. In contrast, our formulation relates the actuated torque or force with resulting state transition. Our approach allows for better generalization and applications in underpowered control domains such as legged locomotion.

Nguyen-Tuong and Peters (2008) proposed similar approach for autonomously modeling the dynamics of robot arms. Their models inferred the inverse kinematics of robot arms and are used in real-time feedback control. They inferred the model using Local GPR (LGPR) that can be inferred and evaluated efficiently. While we also focus on computational efficiency, we extend the use of such efficient models from feedback control to rapid motion planning and real-time re-planning.

Dearden and Demiris (2005) proposed a forward modeling approach based on Gaussian processes to model two motored arms. Their model relates motor commands to the re-

sulting arm motions. However the motor command in their work is binary, while our experiments investigate continuous motor command.

Bongard et al. (2006) inferred the morphology of robots autonomously via an Evolutionary Algorithm (EA) and modeled them in a 3D simulator. The simulated model was used as a surrogate for the real robot and was sufficiently accurate to develop gait. An advantage of their approach was resilience against unexpected damage. However, this work relied on the accuracy of the 3D simulator to develop the model. The design of such 3D physics simulations raises the same fundamental issue of requiring laborious derivations from human engineers. Moreover, since their 3D models require heavy computation, they are not suitable for real-time control.

For goal-oriented control tasks, behavior-based control approaches, such as Reinforcement Learning (Sutton and Barto, 1998) and Neuroevolution (Yao, 1999), are widely studied as alternatives to model-based approaches. These approaches do not rely on models, but instead try to optimize sensorimotor mappings to achieve predefined goals. Although they are successful for achieving given goals, such as inverted cart-pole tasks, they lack the ability to generalize the learned knowledge to other tasks.

## Learning Forward Models

### Forward Modeling

In this work, we seek to find a *forward model* that explains the dynamic relationship between given commands and robot’s state transition. We assume discrete time dynamics in which a robot’s state at time  $t$  is represented as a set of  $m$  sensed values  $X^t = \{x_1^t, \dots, x_m^t\}$ . A forward model for such a dynamics is a function that predicts the state  $X^{t+1}$  at time  $t + 1$  as

$$X^{t+1} \approx f(X^t, C^t), \quad (1)$$

where  $C^t = \{\tau_1^t, \dots, \tau_n^t\}$  is a set of  $n$  command signals at time  $t$ .

An advantage of this formulation is that it can extrapolate the resulting motion of an arbitrary length of motor commands. That is, given initial state  $X^0$  and a command sequence  $C = \{C^0, \dots, C^{T-1}\}$ , resulting state at time

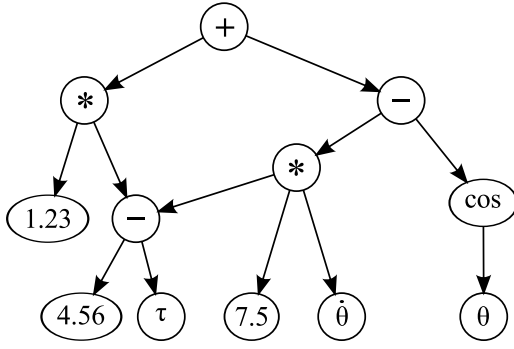


Figure 2: A sample SR model. The DAG representation of a function:

$$f(\tau, \theta, \dot{\theta}) = 1.23 * (4.56 - \tau) + (4.56 - \tau) * 7.5 * \dot{\theta} - \cos(\theta)$$

$t \in [1 : T]$  can be predicted as

$$\begin{aligned} X^t &\approx f(X^{t-1}, C^{t-1}) \\ &\approx f(f(X^{t-2}, C^{t-2}), C^{t-1}) \\ &\approx f(f(\dots f(X^0, C^0) \dots), C^{t-1}). \end{aligned} \quad (2)$$

In this study, all state variables are continuous. The modeling problem is simplified by explicitly predicting second-order differentials of these state variables. We assume that state variables  $X$  can be decomposed as  $X = \langle q, \dot{q} \rangle$ , where  $q$  is a set of linear and angular position variables, and  $\dot{q}$  is a set of their first-order differentials (i.e. velocity variables). We formulate the modeling problem as predicting the function:

$$\ddot{q}^{t+1} \approx g(q^t, \dot{q}^t, C^t),$$

instead of directly inferring function  $f$  in Eq. 1. We calculate state values  $\dot{q}^{t+1}$  and  $q^{t+1}$  via following integration:

$$\begin{aligned} \dot{q}^{t+1} &\approx \dot{q}^t + \ddot{q}^{t+1} \Delta t \\ q^{t+1} &\approx q^t + \dot{q}^{t+1} \Delta t. \end{aligned}$$

The robot's state is represented as a set of angular and linear parameters in most robotic systems. By using generalized state variables, this approach of inferring second-order differential systems can be readily adapted to arbitrary robotic systems. To generate training data set for model inference, random commands are sent to robot's actuators, and  $\langle \ddot{q}^{t+1}, \dot{q}^t, q^t, C^t \rangle$  at each time step is collected as a training data point.

### Learning Models with Symbolic Regression

An SR uses evolutionary algorithm that searches mathematical expressions to explain a given data set. SR has been successfully applied to infer non-linear dynamics, such as conserved laws of nature, accurately (Schmidt and Lipson, 2009). Our work uses SR to search for mathematical expressions that explain the relationship that exists in the training

data. The fundamental idea of SR is to use Genetic Programming (GP) to evolve populations of expressions and selectively generate populations of lower error. Mathematical expressions are represented as Directed Acyclic Graphs (DAGs). GP searches the space of possible graphs and minimizes error by applying genetic operators, such as mutation and crossover. A sample mathematical expression and its DAG representation are shown in Fig. 2. We used Eureka (Schmidt and Lipson, 2009) as SR implementation in our experiments.

## Motion Planning and Control

### Offline Motion Planning

For motion planning, we propose an approach that directly searches the command space via a hill-climbing heuristic, which searches for an optimal command sequence to maximizes a target function. Inferred forward models are used for simulating candidate command sequences in such a function. Optimization algorithm is sketched in Algorithm 1.

**Algorithm 1** Motion planning with forward models using a hill-climbing heuristic

---

```

if Offline planning then
     $C_{\text{best}} \leftarrow$  random commands
     $X^0 \leftarrow$  initial state
else if Real-time planning then
     $C_{\text{best}} \leftarrow$  current motion plan
     $X^0 \leftarrow$  observed state
end if
 $E_{\text{tmp}} = \text{target}(C_{\text{best}}, X^0)$ 
repeat
    for all  $C$  in neighbor( $C_{\text{best}}$ ) do
        if target( $C, X^0$ ) >  $E_{\text{tmp}}$  then
             $C_{\text{best}} = C$ 
             $E_{\text{tmp}} = \text{target}(C, X^0)$ 
        end if
    end for
until Goal condition satisfied or allotted iterations expire
return  $C_{\text{best}}$ 

```

---

Note the evaluation of command sequences is completely dependent on the predicted state transition. Thus, the optimality of the plan depends highly on the accuracy of the predictive model. On the other hand, since the command space is high-dimensional and continuous, the search for the optimal command requires numerous iterations. Therefore, rapid evaluation of target function is vital. Since mathematical expressions are evaluated extremely efficiently on modern computers, SR models can evaluate the function quickly.

### Real-time Motion Control by Re-planning

In practice, the fundamental issue of predicting state transition with forward models is the accumulation of errors

Table 2: Specification of the experiments.

Arm size (H W D)	2 0.4 0.2m
Arm mass	1kg
Gravity	10m s <sup>-2</sup>
Maximum torque for $P_1$	$\tau_{\max}$ : 1.25kg m
Maximum torque for $P_2$	$\tau_{\max}$ : 4kg m
Time step resolution	60Hz
State variables	$X = \{\theta_1, \dot{\theta}_1\} (P_1)$ $X = \{\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2\} (P_2)$ $\theta_1, \theta_2 \in [-\pi : \pi]$
# of neighbors in Algorithm 1	10

over iterative use of prediction as seen in Eq. 2. We resolve this problem using real-time re-planning to adapt accordingly to the errors. As the robot obtains real-time sensor values, it is able to ground the predicted state with the recorded observations. This is implemented by modifying offline planning in accordance to the new observations. In case of re-planning, the hill-climbing algorithm takes the observed state as the initial state and current motion plan as initial command sequence. Since typical robotic systems require high frequency control and feedback, real-time planning must be equally constrained by such critical restrictions. SR models are sufficiently computationally efficient to allow for re-planning, while GRP and SVR models are not. Although re-planning was introduced primarily to adapt to cumulative error, it can start with random motion and plan the motion in purely online fashion.

## Experiments

In this section, we present experimental results. We evaluate our method in the motored single and double pendulum problems, called  $P_1$  and  $P_2$  (Fig. 3). While robots in these problems are mechanically simple, motion planning in this domain remains a challenge for autonomous robotic controllers. Pendulums used in our experiments are under-powered, and thus, achieving most angular positions is a non-trivial task which often requires unexpected motions.

### Experimental Settings

$P_1$  is composed of an arm that is hinged to a stationary point via a motored joint.  $P_2$  has two arms: motor joints connect the first arm to stationary point, while connecting the second arm to the first arm. These pendulums are simulated with Bullet Physics Library (Coumans, 2010), a popular, open-source 3D physics simulator. Detailed specification of the experiment is provided in Table 2.

The goal of the control task in this domain is to move the only arm (of  $P_1$ ) or the upper arm (of  $P_2$ ) to the upright position within allotted time steps (i.e., 600 steps in  $P_1$ , and 1200 steps in  $P_2$ ). Task start with the initial state

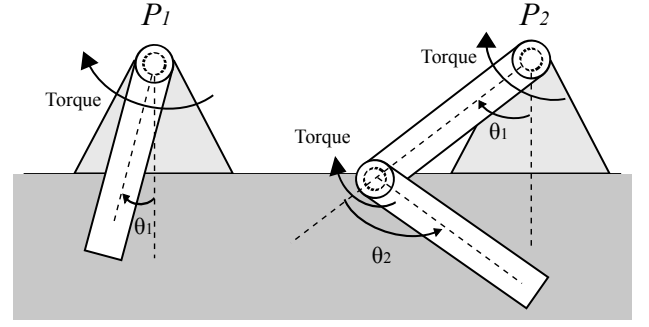


Figure 3: Single ( $P_1$ ) and double ( $P_2$ ) motored pendulum.

of  $X^0 = O$ , where the arms are in the stable equilibrium position. The torque output of each joint motor is limited so that controllers cannot reach to the upright position by simply applying maximum torque in a single direction. Instead, a successful motion requires that the pendulum be swung to accumulate sufficient momentum to eventually achieve the upright position. This additional complexity makes it difficult for existing automated motion controllers to achieve the goal, since evaluating long motor commands via forward models is vital to planning successful motion.

To generate training data set, we actuate a robot using random motor commands. For the pendulum domain, each motor is actuated with a randomly generated torque curve whose torque,  $\tau(t)$ , at time  $t$  is calculated as

$$\tau(t) = \frac{\tau_{\max}}{\sum_{i=1}^N a_i} \sum_{i=1}^N a_i \sin(b_i t + c_i), \quad (3)$$

where  $\tau_{\max}$  denotes maximum torque, and  $a_i$ ,  $b_i$ , and  $c_i$  are random values drawn from uniform distribution under following constraints;

$$\begin{aligned} a_i &\in [0 : 1], \\ b_i &\in [\frac{1}{4} \frac{\pi}{60} : \frac{\pi}{60}], \\ c_i &\in [0 : 2\pi]. \end{aligned}$$

This formulation results in a composite wave of  $N$  individual sine waves. We set  $N = 3$  throughout the experiments. Random actuation lasts for 1 minute, resulting in 3600 time steps. In the double pendulum domain, two distinct torque curves are generated with different random seeds.

### Model Inference and Cross-validation Evaluation

Given the training data, we can learn forward models using off-the-shelf regression algorithms. We compare Symbolic Regression (SR) with Support Vector Regression (SVR) and Gaussian Process Regression (GPR). We used Eureqa (Schmidt and Lipson, 2009), libSVM (Chang and Lin, 2001), and Weka (Hall et al., 2009) libraries for SR, SVR, and GPR, respectively.

Table 3: Comparison of regression algorithms. Listed numbers denote correlation coefficients of inferred models. Computation time for inferring each model is shown in parenthesis.

Target	GPR	SVR	SR (short)	SR
$\ddot{\theta}_1$ in $P_1$	0.9996 ( $\approx 4\text{min}$ )	0.9733 ( $< 1\text{sec}$ )	<b>1</b> ( $= 4\text{min}$ )	<b>1</b> ( $= 2\text{hours}$ )
$\dot{\theta}_1$ in $P_2$	0.7063 ( $\approx 4\text{min}$ )	0.9588 ( $\approx 18\text{sec}$ )	0.9639 ( $= 4\text{min}$ )	<b>0.9700</b> ( $= 8\text{hours}$ )
$\ddot{\theta}_2$ in $P_2$	0.7455 ( $\approx 4\text{min}$ )	0.9286 ( $\approx 27\text{sec}$ )	0.9080 ( $= 4\text{min}$ )	<b>0.9751</b> ( $= 8\text{hours}$ )

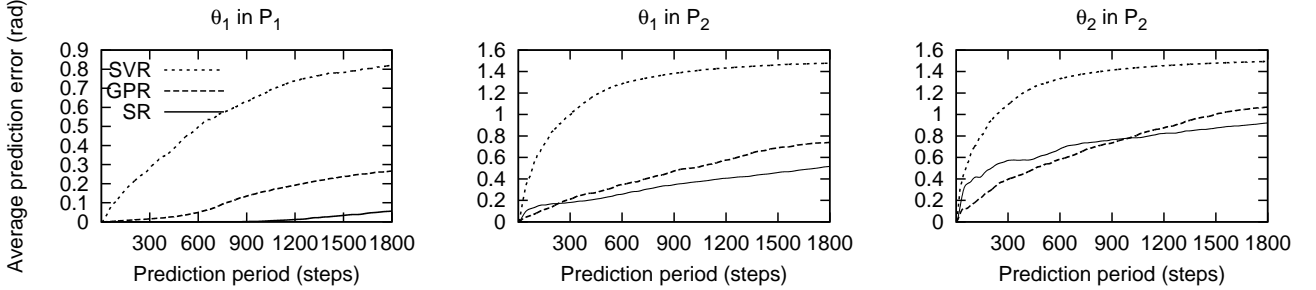


Figure 4: Average error on varying prediction periods.

The model inference results of all approaches were compared on cross-validation data sets and performance is measured with correlation coefficient. The results are summarized in Table 3. Computation time for inferring models on Intel Core2 Duo 3.06GHz are also listed in the table. Since SR is a stochastic process, longer training periods may yield better results, while GPR and SVM are deterministic algorithms that do not improve with additional time. We tested on short, being matched with the time GPR inference took, and long training period for SR. Since model inference is an offline process, time constraints are typically not strict.

The results indicate that SR is superior to both GPR and SVR, given long training time. SR inferred a virtually perfect model for  $\theta_1$  in  $P_1$ . Even with shorter training period, SR models are comparable to those inferred with GPR or SVR. In the following experiments, we use SR models inferred with longer training period.

### Model Accuracy on Novel Command Sequences

Since cross-validation results on training data sets do not necessarily reflect generalization performance of inferred models on novel data sets, we used additional tests to inspect and analyze differences in these models. To evaluate models on novel data sets, we generated 20 random torque curves using Eq. 3 with different random seeds and predicted resulting motion using the inferred models. Their prediction error was evaluated as the difference from actual motion. Since one of our concerns is the effect of cumulative errors, we vary the prediction period from 1 to 1800 steps to see how each model behaves during iterative predictions. Average predictive error on each joint angle over the prediction period is shown in Fig. 4.

We can readily see that average prediction error tends to

increase, as the prediction period gets longer. This implies that the cumulative error harms predictive performance over iterations. For predicting  $\theta_1$  in  $P_1$ , it is clear that the error is less pronounced in the SR models, suggesting that the symbolic representation has a more consistent model representation. SVR did poor job on novel data sets, in spite of good performance on cross-validation. The differences of performance among three models are statistically significant ( $p < 0.05$ ) for prediction periods of 60, 120, 600, and 1200 steps.

In  $P_2$ , SVR again marked poor performance, for predicting both  $\theta_1$  and  $\theta_2$ . GPR and SR performed comparably. While GPR models outperformed SR models for short periods (i.e., lower than 233 and 997 steps for  $\theta_1$  and  $\theta_2$ , respectively), SR was superior for long periods. This implies that SR models would be more robust to cumulative error over iterative predictions. Another implication is that cross-validation results, in which GPR models are evaluated badly, do not reflect the models' generalization performance appropriately.

### Evaluation of Offline Motion Planning

Given inferred dynamics models, the motion planning is formulated as an optimization problem. The target function is defined as follows:

$$\text{target}(\theta_1) = \max_{\theta_1^t \in \theta_1} |\theta_1^t|$$

where  $\theta_1 = \{\theta_1^0, \dots, \theta_1^T\}$ , and  $\theta_1^t$  is the angle of the target joint at time  $t$ . Only the initial state is provided to the models. Given candidate command sequence  $C$  and initial state variables  $X^0$ , the algorithm simulate entire state transition over  $T$  steps. Our method plans the motion by searching for

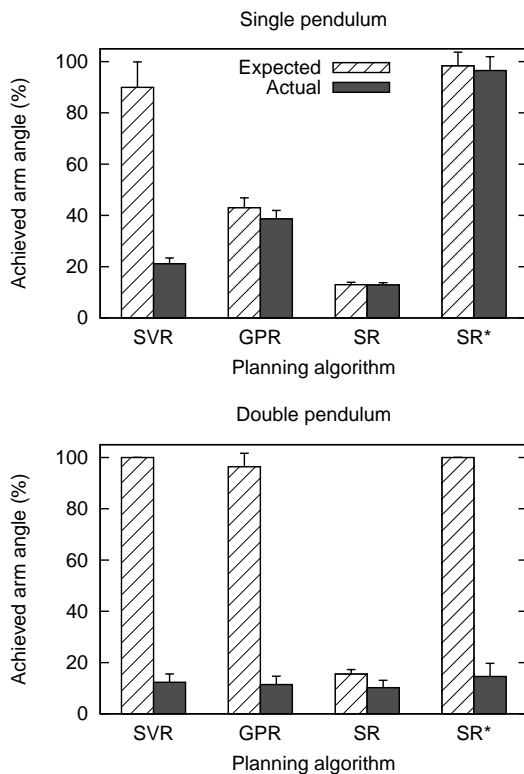


Figure 5: The performance on motion planning tasks.

a command sequence whose simulated motion maximizes this function.

To evaluate motion planning performance of each model, the hill-climbing approach, described in Algorithm 1, was applied for 20 runs per each. The planned commands were executed on actual robot, and their actual performance were evaluated based on Eq. 4. We compared the actual performance with the expected performance to analyze the discrepancy between predicted and actual performance.

In initial tests, the hill-climbing approach for all models was executed for 500 iterations. However, there is a significant difference in the computational efficiency of SR, GPR and SVR, with SR executing at least three orders of magnitude faster than GPR and SVR in both  $P_1$  and  $P_2$  experiments. To provide comparable results with respect to computational effort, an additional SR, called SR\*, experiment was implemented with 500000 iterations.

The performance of the SVR, GPR, SR and SR\* motion planning is shown in Fig. 5. It is clear that SVR and GPR tends to overestimate its capability, often allowing for highly optimized plans that do not translate to real world results. In comparison, SR provides a truer prediction of the real world even if the hill-climbing problem becomes subsequently harder. When normalized for computational effort, SR\* does a significantly better job, compared to SVR and GPR.

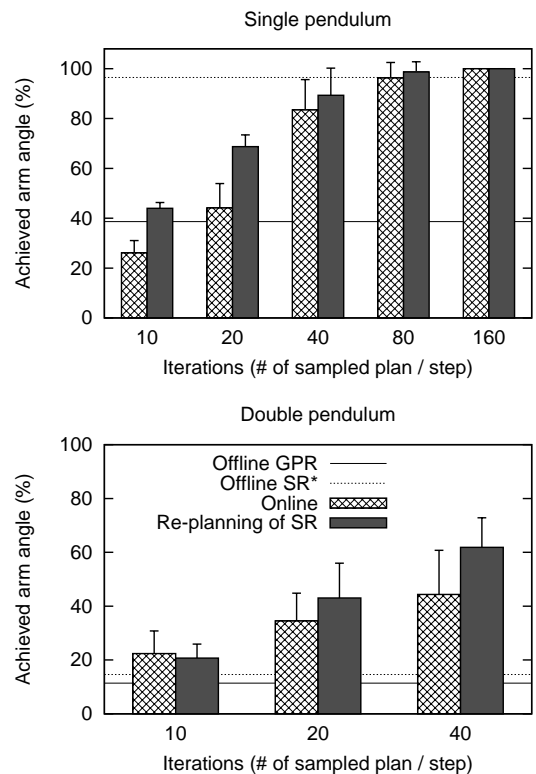


Figure 6: The performance of real-time planning. The performances of offline methods are given as baselines.

In the double pendulum problem, SVR and GPR predicted a nearly optimal solution with certainty, but in fact, the actual performance was poor. SR achieved similar performance, but had a more realistic prediction. Although SR\* had the best actual performance, its performance was far poorer than expected.

## Evaluation of Real-time Motion Control

This section provides the comparison for the real-time re-planning motion control. We experimented two re-planning approaches: first approach started with an initially optimized plan (i.e., re-planning of the plan generated by SR that is described in the last section); while second one started with a random plan, resulting in purely online planning and control. These two are compared with the offline GPR planning and SR\* planning. For the re-planning approaches, the number of iterations per step (IPS) was doubled until the real time constraint could not be achieved. This limit was resulted 160 IPS and 40 IPS for the single and double pendulum, respectively. Since SVR and GPR models could not achieve even 1 IPS for both problems, they are not applied to re-planning. The results are shown in Figure 6.

The performance of online and “re-planning of SR” approaches improve with increased IPS, significantly outperforming offline GPR and SR\* ( $p < 0.05$ ) with maximum



IPS (i.e., 160 for single and 40 for double pendulum). “Re-planning of SR” typically showed remarkable improvement over the purely online approach, resulting in the best performance among all tested approaches with maximum IPS. When the IPS is set to maximum, the total number of iterations required for “re-planning of SR” approach was 96500 iterations and 48500 iterations for single and double pendulum problems, respectively. While offline SR\* consumed 500000 iterations, its performance is poorer than that of “re-planning of SR”.

In summary, the benefits of SR modeling over GPR and SVR modeling are two-fold: first, it provides more accurate prediction that is vital to planning; and second, the reduced computational effort makes efficient real-time applications possible.

## Conclusion and Future Work

In this paper, an SR-based method was proposed to model a robot’s dynamics autonomously and the inferred model was used for motion planning and control tasks. The model is represented as a mathematical expression that accurately explains the robot’s dynamics and allows for fast computation. These features enable fast motion planning and real-time control based on re-planning, resulting in significantly superior control performance over GPR and SVR-based methods.

Possible applications of our method include forward modeling and motion planning for the locomotion of legged robots or soft robots. Soft robots are an area of particular interest since deriving a dynamic model is a difficult and daunting task. However, forward models of these robots are considered significantly more complex than of pendulums. Since the difficulty of model inference via SR is known to increase dramatically as the complexity of the true model increases (Schmidt and Lipson, 2008), we will need to develop novel techniques to infer models of more complex dynamics.

This paper is an initial investigation of applying SR for robotic modeling and there is room for further optimization. Although this work used a simplistic hill-climbing heuristic approach for motion planning, more complex and superior algorithms can be applied instead. It will also be possible to use SR to infer inverse models and utilize them in feedback control. We believe that the coexistence of accuracy and efficiency in SR models will help a novel class of algorithms in robotics to emerge.

## Acknowledgements

The authors thank Daniel Ly of the Creative Machines Laboratory at Cornell University for his invaluable comments on the present paper. This work was supported in part by NIH NIDA grant RC2 DA028981, NSF CDI Grant ECCS 0941561. The content of this paper is solely the responsibility of the authors and does not necessarily represent

the official views of the sponsoring organizations. Hirotaka Moriguchi is supported by a scholarship from Funai Foundation for Information Technology.

## References

- Bongard, J., Zykov, V., and Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science*, 314(5802):1118–1121.
- Chang, C. and Lin, C. (2001). *LIBSVM: a library for support vector machines*. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Coumans, E. (2010). *Bullet 2.76 Physics SDK Manual*. Software available at <http://bulletphysics.org/>.
- Dearden, A. and Demiris, Y. (2005). Learning forward models for robots. In *Proceedings of IJCAI*, pages 1440–1445.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. (2009). The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.
- Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology*, 9(6):718–727.
- Koza, J. (1992). *Genetic programming: on the programming of computers by means of natural selection (Complex Adaptive Systems)*. The MIT Press.
- Nguyen-Tuong, D. and Peters, J. (2008). Local gaussian process regression for real-time model-based robot control. In *Proceedings of IROS*, pages 380–385.
- Schmidt, M. and Lipson, H. (2008). Coevolution of fitness predictors. *IEEE Transaction on Evolutionary Computation*, 12(6):736–749.
- Schmidt, M. and Lipson, H. (2009). Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85.
- Sturm, J., Plagemann, C., and Burgard, W. (2008). Unsupervised body scheme learning through self-perception. In *Proceedings of ICRA*, pages 3328–3333.
- Sutton, R. and Barto, A. (1998). *Reinforcement learning: an introduction*. The MIT Press.
- Wolpert, D., Ghahramani, Z., and Jordan, M. (1995). An internal model for sensorimotor integration. *Science*, 269(5232):1880–1882.
- Yao, X. (1999). Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447.

# Identifying Species by Genetic Clustering

Jaimie Murdock<sup>1</sup> and Larry S. Yaeger<sup>1</sup>

<sup>1</sup>School of Informatics & Computing, Indiana University, Bloomington, IN 47408  
jammurdo@indiana.edu, larryy@indiana.edu

## Abstract

Complex artificial life simulations can yield substantially distinct populations of agents corresponding to different adaptations to a common environment or specialized adaptations to different environments. Here we show how a standard clustering algorithm applied to the artificial genomes of such agents can be used to discover and characterize these subpopulations. As gene changes propagate throughout the population, new subpopulations are produced, which show up as new clusters. Cluster centroids allow us to characterize these different subpopulations and identify their distinct adaptation mechanisms. We suggest these subpopulations may reasonably be thought of as *species*, even if the simulation software allows interbreeding between members of the different subpopulations, and provide evidence of both sympatric and allopatric speciation in the Polyworld artificial life system. Analyzing intra- and inter-cluster fecundity differences and offspring production rates suggests that speciation is being promoted by a combination of post-zygotic selection (lower fitness of hybrid offspring) and pre-zygotic selection (assortative mating), which may be fostered by reinforcement (the Wallace effect).

## Introduction

Artificial life simulations exhibit complex agent-based behaviors, which persist and evolve through genetic recombination and mutation. Unless explicit speciation is built into the simulation, identifying emergent species in these simulations is difficult, both theoretically and practically. Here we demonstrate a technique for identifying subpopulations of agents using a clustering algorithm to identify groups of agents with shared genetic attributes. The resulting clusters might reasonably be considered distinct species, and allow us to identify some of the different adaptation mechanisms adopted in the simulation. Examining the temporal distribution of these clusters allows us to better understand the evolutionary course of speciation and adaptation in our simulations, and may offer some insights into speciation in biological ecosystems.

Understanding speciation is one of the key problems in biology. Much debate centers around the role of *allopatric* (geographically isolated) vs *sympatric* (shared environment) species divergence. The significance and driving forces

of sympatric speciation have been controversial since the ideas were introduced by Wallace (1899) and championed by Dobzhansky (1937). *Disruptive selection* (adaptation to distinct fitness peaks) in combination with *reinforcement* (the selection pressure that results from reduced fitness of hybrids; aka the *Wallace effect*) leads to *assortative mating* (a preference for related partners) thus providing a basis for sympatric speciation. Despite the simplicity and attractiveness of these ideas, the so-called Modern Synthesis largely discarded the idea of selective speciation, instead attributing divergence to more readily observable geographic isolation (Mayr and Provine, 1998), and a variety of models (reviewed in (Kirkpatrick and Ravigné, 2001)) have led many to conclude that sympatric speciation, while possible, will only be found under very limited circumstances (Felsenstein, 1981). However, though the jury is still out, empirical evidence for reinforcement driving sympatric speciation does exist ((Sætre et al., 1997; Ortiz-Barrientos et al., 2004; Silvertown et al., 2005) and others) and recent theoretical and modeling work have suggested potential mechanisms (such as competition overwhelming selection towards a single method of resource utilization) for overcoming the perceived limitations on sympatry (Dieckmann and Doebeli, 1999; Kondrashov and Kondrashov, 1999; Van Doorn and Weissing, 2001). For high-level reviews see (Butlin and Tregenza, 1997; Tregenza and Butlin, 1999; Weissing et al., 2011). In this work, both *pre-zygotic* (pre-mating) and *post-zygotic* (post-mating) selection are observed, suggesting reinforcement may be playing a role in our speciation events—both sympatric and allopatric followed by population mixing.

In the life sciences clustering algorithms are applied in many areas, including the analysis of clinical information, phylogeny, genomics, and proteomics (Zhao and Karypis, 2005). Mallet (1995) proposed gene clustering as a preferred method for the rigorous identification of biological species (as opposed to taxonomic features). We seek to import these concepts and tools from the realm of biology into our artificial life work to help us better understand the evolutionary dynamics of our model ecosystem, though we believe there may be some general principles that apply to both artificial

and natural ecosystems.

The use of gene clustering for speciation has been explored in genetic algorithms by Hocaoglu and Sanderson (1995) and in computational ecosystems by Aspinall and Gras (2010). The Aspinall and Gras predator-prey simulator has some traits in common with ours, but defines two distinct agent classes which do not interbreed, and the clustering analysis is performed during the simulation and allowed to control reproductive success, thus allowing it to drive the speciation process. By contrast, there is no impact of cluster membership or genetic distance on reproductive success in the work reported here, and all gene clustering analysis is performed *post hoc*, after a simulation has run its course.

Clustering algorithms (reviewed in Hartigan (1975); Kaufman and Rousseeuw (2005)) rely upon two key elements: the distance function used to measure object similarity and the algorithm used to partition the data. The distance function must account for the “curse of dimensionality” (Bellman, 1957) intrinsic to high dimensional spaces in general and evolutionary algorithms employing large, high-dimensional genomes in particular. Clustering algorithms with a pre-specified number of clusters—such as k-means clustering (MacQueen, 1967)—though widely used, suffer from the simple fact that the number of clusters may not be known *a priori*.

Information theory (Shannon, 1948) allows us to partially alleviate the curse of dimensionality. Through the process of variation and selection those genetic dimensions which most affect an agent’s fitness will be selected for and conserved, thus exhibiting low entropy across the population of agents, while those which are inconsequential will descend into a random distribution. By weighting genetic dimensions with *certainty* (i.e.,  $1 - \text{entropy}$ ) those genetic features most significant to the agents’ survival and reproduction will be emphasized during the partitioning into clusters, while spurious proximity in the inconsequential dimensions is ignored.

Algorithmically, we have chosen to use the QT (Quality Threshold) Clustering algorithm (Heyer et al., 1999; Scharl and Leisch, 2006), which clusters based on a maximum intra-cluster distance (diameter), rather than a set number of clusters.

## The Artificial Life Software

This research was carried out using Polyworld (Yaeger, 1994), a computational ecology with a long history, in which populations of haploid agents evolve, each possessing a suite of primitive behaviors (move, turn, eat, mate, attack, light, focus) under continuous control of an Artificial Neural Network (ANN) employing (in this case) discrete-time, firing-rate neurons with synapses that adapt via Hebbian learning. The wiring diagram of the ANN is encoded in the organism’s genome, via a statistical description of the number of neural groups of excitatory and inhibitory neurons, synaptic connection densities, regularity of connections, and learning

rates. The only epistatic interaction between genes derives from the role played by the genes expressing the number of neural groups and the number of neurons in each group in controlling whether the corresponding inter-group and inter-neuron connections are expressed. For a detailed discussion of Polyworld’s genetic encoding scheme, see (Yaeger, 1994).

Input to the ANN consists of pixels from a rendering of the scene from each agent’s point of view. Output from the ANN consists of the aforementioned primitive behaviors. For the simulation discussed here, there are 2,486 genes devoted to specifying the neural topologies (but not synaptic weights) of ANNs with up to 217 neurons and 45,854 synaptic connections. The actual neuron count ranged from 14 to 163, with a mean of 48, and the synapse count ranged from 46 to 9,034, with a mean of 656. A small number of genes (8) characterize the agents’ simple morphologies, metabolisms, and meta-genetics, in terms of size, strength, maximum speed, fraction of energy contributed to offspring, ID (green color component), mutation rate, number of crossover points, and lifespan. Thus there are 2,494 genes in all used in the clustering process.

All actions of the agents consume energy, so they must replenish their energy levels by seeking out and consuming food or by killing and eating other agents. Normally there are also per-neuron and per-synapse energy costs, but for consistency with some evolution-of-complexity experiments these were disabled for the results reported here. Reproduction occurs when two collocated agents simultaneously express their mating behaviors.

The simulation is initially seeded with a uniform population of agents that have the minimum number of neural groups and a nearly minimal number of neurons and synapses. While predisposed to some potentially beneficial behaviors, such as running towards food (green) and away from aggression (red; see (Yaeger, 1994) for details on color use in Polyworld), these seed organisms are not a viable species. I.e., without evolution they cannot sustain their numbers through their reproductive behaviors and will inevitably die out.

For these analyses the world was configured as in (Yaeger et al., 2008), with two barriers running 90% of the depth of the world, but left open for the remaining 10% of the world, so populations are able to mix relatively easily, but not with complete freedom. 80% of the food is grown in a patch occupying 40% of world depth at the open end of the barriers, 20% in a patch occupying 10% of world depth at the closed end of the barriers. This layout may be seen in Figure 1.

As simulations progress both the structural architecture of the ANNs and the activation of every neuron at every time step for every agent may be recorded, thus permitting investigation into evolutionary trends in network structure and function (Yaeger et al., 2010). Agent genomes may also be recorded, and these recorded genomes serve as the basis for

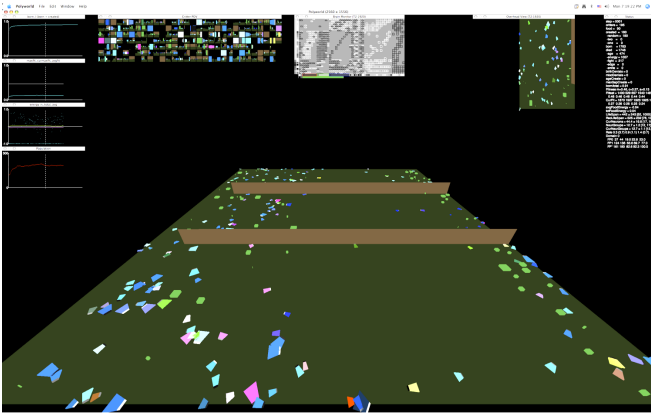


Figure 1: Polyworld simulation environment

the clustering analysis described here. Some genes exhibit smooth, general trends over the course of the simulation, but others demonstrate short, sharp changes that correspond to temporal cluster boundaries, as will be discussed later.

### The Clustering Algorithm

The clustering task can be divided into two subproblems: the distance function used to measure object similarity and the clustering algorithm used to partition objects. For the distance function, we used entropy-weighted Euclidean distance over each agent's genome. For the clustering algorithm, we used a variation of the QT-Clust algorithm (Heyer et al., 1999; Scharl and Leisch, 2006), with the addition of a new algorithmic improvement to allow for multiple cluster selection on each pass and a precalculation of point-wise distances for greater efficiency.

### The Distance Function

Genomic data in artificial life simulations are afflicted by the curse of dimensionality (Bellman, 1957), and the current Polyworld genome consists of nearly 2,500 genes! Fortunately, the process of selection in evolutionary algorithms gives a way to identify genes which are likely to differentiate subpopulations. Genes with a high impact on agent fitness will be selected for and conserved, while those which are inconsequential will trend towards a random distribution. By taking the information certainty (1 - Shannon Entropy) of each gene, the relative importance of each dimension may be used to weight the many dimensions:

$$H(g) = - \sum_{i=0}^{N_s} p(g_i) \log_2(p(g_i))$$

$$certainty(g) = 1 - H(g)$$

where  $g$  is a specific gene, the  $g_i$  are the gene values (states), and  $N_s$  is the number of possible gene states. Probabilities were calculated for 16 bins of 16 gene values, capturing the

full range of these 8-bit genes (0-255), over the entire population of 29,564 agents extent during the full evolutionary simulation.

While each gene of the Polyworld genome is specified by an 8-bit value, the full range of genetic values may not be expressed over the course of a simulation. In comparing genomic data, the difference along this distribution is more important than the raw score. To address this issue when calculating genetic distances between agents, we have normalized the measure of each gene dimension, by calculating the genes' z-scores:

$$z(x) = \frac{x - \mu}{\sigma}$$

where  $x$  is the raw gene value,  $\mu$  is the mean value of that gene, and  $\sigma$  is the standard deviation of that gene's values).

After normalizing gene values to produce gene z-scores, distances are calculated between z-scores, weighting the relative importance of each gene by its certainty. Our distance metric is therefore the certainty-weighted squared-Euclidean distance of z-scores:

$$dist(x, y) = \sum_{i=0}^{N_g} (w_i(z(x_i) - z(y_i)))^2$$

where  $x$  and  $y$  correspond to two agents and their genomes,  $N_g$  is the total number of genes in the genome,  $w_i$  is the certainty calculated for each specific gene  $i$ , and  $z(x_i)$  and  $z(y_i)$  are the z-scores of gene  $i$  in the genomes of agents  $x$  and  $y$ .

### The QT-Clust Algorithm

Clustering algorithms rely upon the fixation of one or more variables: number of clusters, similarity of elements in the cluster, or number of elements in each cluster. Effective clusters should maximize inter-cluster distances, while minimizing intra-cluster distances (cluster diameter). Traditional k-nearest-neighbor approaches (MacQueen, 1967) require the number of clusters to be specified *a priori*. Additionally, these algorithms encounter the hubness phenomenon in which a centroid may be a common nearest-neighbor in Euclidean space, building large diameter clusters. This phenomenon is exacerbated by high-dimensionality (Beyer et al., 1999; Radovanović et al., 2010).

To avoid these issues, we have opted to use the QT-Clust algorithm (Heyer et al., 1999; Scharl and Leisch, 2006), which is a nearest-neighbor clustering approach fixing cluster diameter ( $\epsilon$ ), rather than the number of clusters. This algorithm is particularly well suited for data discovery problems, such as gene analysis (the original use case). Adjustment of the cluster diameter parameter provides a means of controlling cluster fit that is both more intuitive and practical than algorithms requiring explicit specification of the number of clusters. (E.g., we are unlikely to have chosen

---

**Algorithm 1: QT-Clust**

---

**Input:**  $G, \epsilon$ **Output:** Clusters

```
if  $|G| \leq 1$  then
  output  $G$ 
else
  // Cluster building
  foreach  $i \in G$  do
     $flag := TRUE; C_i := i;$ 
    while  $flag$  and  $C_i \neq G$  do
      find  $j \in G - C_i : diameter(C_i \cup j)$  is min;
      if  $diameter(C_i \cup j) > \epsilon$  then
         $flag := FALSE$ 
      else
         $C_i = C_i \cup j$ 
  // Cluster selection
   $C := C_0 \dots C_{|G|};$ 
  while  $|C| > 0$  do
    identify set  $P \in C$  with max cardinality;
     $G := G - P;$ 
     $C := \{X \in C : |X \cap P| = 0\};$ 
    output  $P;$ 
  QT-Clust( $G, \epsilon$ )
```

---

values of 8, 29, and 108 for the number of clusters we ended up focusing our attention on, but specifying cluster diameter in terms of standard deviations that produced these clusterings seemed reasonably natural.) The iterative approach used by QT-Clust also avoids issues of hubness common to nearest-neighbor clustering algorithms by creating an  $\epsilon$ -neighborhood graph around each agent. The largest of these groupings is then selected and removed from the population to be re-clustered, thus eliminating the effect of outliers and hubs (Radovanović et al., 2010).

The algorithm has two stages. First, a cluster is built starting with each agent within the population ( $G$ ). The cluster is built by adding the next closest agent to the cluster, until a threshold ( $\epsilon$ ) of maximum distance is reached. Cluster construction may be done in parallel for a significant speed increase. Then, each of these clusters is passed through a filtering step, which selects the largest candidate that does not overlap with a previously selected cluster, until no viable candidates remain. This multiple selection amortizes the time complexity of the original QT-Clust algorithm, while maintaining its quality control advantages. After filtering, unclustered elements are then reclustered within the remaining population until all elements are classified.

## Results

We ran this algorithm on Polyworld simulation data containing 29,564 agents (distributed over 30,000 time steps), contained in 1.9GB of genomic data. Simulation paramet-

$\epsilon$	1.5	1.75	2	2.125	2.25	2.5	2.75
# clust	2063	750	108	29	8	3	3

Table 1: Resulting cluster counts for different  $\epsilon$  thresholds

ters are identical to those presented in previous work on the evolution of neural complexity (Yaeger et al., 2008). While previous work has focused on general trends, combining the results of multiple runs and applying standard tests of statistical significance, here we wish to tease apart the dynamics of a particular simulation, and we are interested in the degree to which cluster analysis and a species/sub-population perspective can inform the understanding of those dynamics. We would expect the details of cluster/species formation to vary from run to run, even when nothing changes but the pseudo-random number generator’s seed, and have seen hints of such variation in previous work on complexity trends.

For the discussion below, we define  $\epsilon$  as a factor of the sum of all certainty weightings:

$$\epsilon(x) = x \sum_{i=0}^{N_g} w_i$$

This sum is equivalent to the weighted distance between two genomes which differ by 1 standard deviation on each dimension, due to z-score normalization. Thresholds were set between 1.5 and 3 times the sum of the certainty values, at increments of .25.

## Behavior Across Different Thresholds

Table 1 shows the number of clusters identified for varying levels of  $\epsilon$ . Figure 2a-c show the population of each cluster over time for  $\epsilon = 2.0, 2.125, 2.25$ . The progression from a large diameter to a smaller diameter shows each cluster splintering. Whether these show hierarchical clustering is a question for further empirical study.

## Temporal Trends in Clusters

Figure 2b shows that while the larger clusters tend to be replaced serially over time, other, smaller clusters emerge, coexist with one or more of the larger clusters for extended periods of time, and are ultimately extinguished, suggesting the emergence, persistence, and decline of subordinate species. This also suggests we may be seeing reproductive isolation of sub-populations, despite the fact that Polyworld does not in any way inhibit cross-cluster reproduction. This could be due to pre-zygotic, assortative mating preferences (unpublished work suggests agents attend to both genetically and behaviorally determined color expressions) or to post-zygotic disruptive selection effects in a Dobzhansky-Muller manner—hybrid offspring expressing neural architectures that are sub-optimal in themselves, or in combination with “physiological” characteristics that affect energy requirements. We look at both possibilities below.

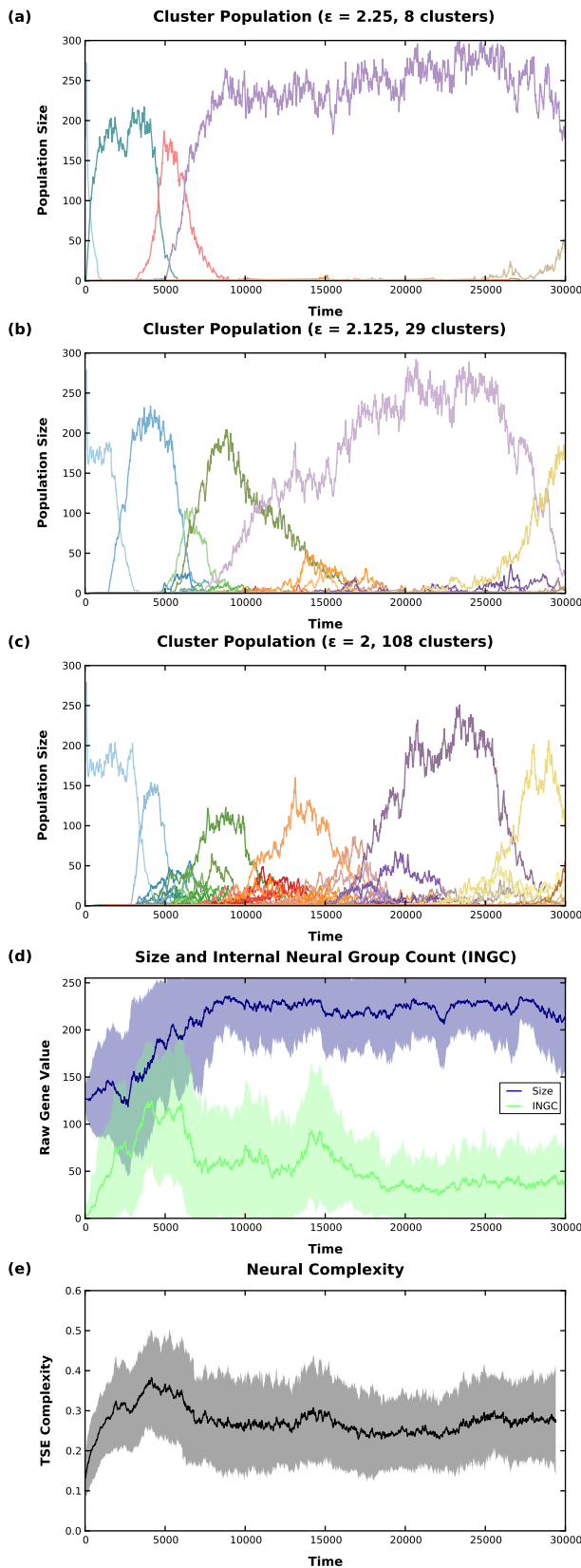


Figure 2: Temporal trends in cluster populations for  $\epsilon = \{2.0, 2.125, 2.25\}$ , two high-certainty genes (size and internal-neural-group count) exhibiting different selection behaviors, and TSE complexity. Genes and complexity shown as population means with standard deviation bands.

	Size	Start	Temporal Peak	End	Neural Complexity	Genetic Size	Genetic INGC
0	1062	0	78	3749	0.2445	133.6	35.1
1	2278	1311	4087	8441	0.3626	172.4	101.0
5	769	4408	6694	11585	0.3657	202.5	98.3
9	3983	5119	8772	17509	0.3058	224.7	62.3
16	205	9168	14722	20192	0.3563	221.3	106.3
17	767	8795	13813	27611	0.3257	215.8	77.1
21	16732	6394	20672	30000	0.2876	225.4	41.9
23	397	11487	28572	30000	0.2861	200.5	38.8
24	273	13207	26594	30000	0.2619	185.3	29.1
27	2202	15126	29565	30000	0.3114	222.0	45.7

Table 2: Raw data from QT-Clust with  $\epsilon = 2.125$ . Shown are the origin, peak, and extinction of each major cluster, the TSE complexity, and mean values of the size and internal-neural-group-count (INGC) genes. Gene values are in the raw 0–255 range. Clusters with  $< 700$  members appear in light gray. Clusters with  $< 200$  members are not shown.

For the larger clusters, from cluster populations alone we cannot distinguish between roughly monotonic, anagenetic (within lineage) changes and true cladogenetic (divergent) speciation. However, long periods of temporal overlap during transitions suggest we may be seeing true speciation in large clusters as well, as distinct, new clusters emerge and are simply more successful than either the short-lived small clusters or the previous large cluster.

### Temporal Trends in Genes

The use of clusters allows us to identify genetic differences between different subpopulations, including temporal trends in specific genes known to distinguish different subpopulations. Figure 2d shows different selection patterns for two high-certainty genes positioned below the cluster population graphs to allow comparison of their temporal trends. Table 2 shows the corresponding raw data for all clusters with a population size greater than 200.

The size gene (*certainty* = 0.3515) shows a nearly monotonic selection pattern. Only the initial seed population has a relatively small size. By the time of the transition from the second to the third major cluster, size has reached the level at which it will plateau—around 220. By contrast, the internal-neural-group-count gene (*certainty* = 0.2058) shows a more variable selection pattern, which corresponds to trends in neural complexity as discussed below. These changes also correspond to cluster emergence and decline, as discussed in Cluster Characterization.

### Neural Complexity

Tononi-Sporns-Edelman neural complexity (TSE complexity) (Tononi et al., 1994) gives an indication of the neural structure and function for each agent. Figure 2e shows the mean TSE complexity over time for the simulation being analyzed. In a past study, complexity was shown to be highly selected for only during periods of behavioral adaptation of the agents to their environment (Yaeger, 2009), in keeping with the tautology of evolutionary selection applying only when the subject of selection confers an evolutionary ad-

Clusters	children	grandchildren	child-rate	grandchild-rate
Same	2.04 (0.02)	4.04 (0.05)	6.54 (0.06)	10.6 (0.2)
Diff	1.89 (0.00)	3.57 (0.03)	5.11 (0.12)	7.76 (0.3)

Table 3: Reproductive success—numbers of offspring from parents of the same or different clusters and child-production rates per 1,000 contacts with agents from same or different clusters (stderr in parens), using  $\epsilon = 2.125$ .

vantage. The current results are in general agreement with previous simulations, showing strong selection for complexity in early populations during the period in which they are evolving to adopt an Ideal Free Distribution (Fretwell and Lucas, 1970; Fretwell, 1972) of agents to the heterogeneous resources of the simulated environment, plateauing around step 7500, and followed by a long stretch of relative stability lasting for the rest of the simulation. However, we see here a bump in complexity around  $t=15,000$ , unique to this particular simulation, that our clustering analysis reveals to be the result of a corresponding bump in internal-neural-group count deriving from the emergence and decline of a pair of specific sub-populations (clusters 16 and 17).

## Discussion and Conclusions

Whether discussing the larger clusters, that replaced each other somewhat serially, or smaller clusters that represented sub-populations coexistent with the larger populations, we think it may be reasonable to conceive of these clusters as *species* within our artificial simulation. Since the simulation does not explicitly prevent interbreeding between clusters or base reproductive success on genetic distance, perhaps they should be considered *proto-species*, but the fall and rise of sub-populations, with significantly different genetic makeup from the dominant population, suggests a degree of specificity and persistence of species identity. Even the dominant populations may demonstrate speciation and competition between species, given the degree to which they overlap in time; e.g., note in Figure 2b that the cluster rising to dominance at the end of the run (light orange – cluster 27) first appeared barely over half way through the simulation ( $t=15,126$ ) well before the previous dominating population (light purple – cluster 21) had reached its peak population ( $t=20,672$ ). This occurs despite a relatively simple environment in which agents are free to mix and in which there is only one kind of energy resource (two if you distinguish between food that is grown and food derived from the carcasses of agents that are killed).

As Mallet (1995) notes, “Clusters can remain distinct under relatively high levels of gene flow provided there is strong selection against intermediates; species will be maintained when selection balances gene flow.” Lacking geographic isolation, sympatric speciation is typically thought to require disruptive selection to elicit distinct phenotypes and genotypes, coupled with selection for assortative mating to elicit reproductive isolation.

If disruptive selection and poor hybrid fitness are playing a role in balancing gene flow, we should see differences in the fitness, as measured by fecundity, of offspring from parents belonging to the same or to different clusters. To investigate this hypothesis we examined the number of children and the number of grandchildren produced by agents born to parents from the same or from different clusters. The left-hand columns of Table 3 summarize the results. Though the differences are modest, the offspring of parents from the same cluster produce more offspring than do the offspring of parents from different clusters, and those offspring are themselves more fecund. The magnitude of the differences are about 10x the standard error rates observed in the population (shown in parentheses), thus there is at least a modest post-zygotic selection pressure at work. Amplified across multiple generations it is easy to see how intra-cluster breeders will outperform inter-cluster breeders and produce ever more distinct sub-populations—species—even sympatrically. This is basically the first half of Wallace and Dobzhansky’s proposed route to sympatric speciation.

If reinforcement is producing pre-zygotic selection and assortative mating, we should see differences in the rate at which agents produce offspring when they come in contact with agents from the same or different clusters. To investigate this possibility we examined the number of children and grandchildren produced *per contact* with other agents from the same or different clusters. For this analysis it is important to normalize birth rates by contact counts, since any kind of temporal, behavioral, or geographical isolation can and does significantly skew the number of potential reproductive encounters between same and different clusters for a given agent. The right-hand columns of Table 3 summarize these results. Both the child- and grandchild-production rates (per 1,000 contacts) are greater for encounters with agents from the same cluster than for agents from a different cluster. Here again, though the magnitude of the differences is small, they are roughly 10x the standard error rates observed in the population. Thus there is at least a weak pre-zygotic selection pressure at work.

Certain characteristics of the current simulated environment—especially the partial barriers, that are a holdover from previous experiments looking at the evolution of complexity—make it difficult to entirely tease apart sympatric vs allopatric speciation. In movies showing cluster membership over time we see clusters emerge and persist alongside existing clusters in a fully sympatric fashion. But we also see evidence of allopatric speciation, with new clusters emerging in and coming to dominate one food patch before spreading to the other—in fact, having difficulty invading the second food patch. So we currently believe both forms of speciation are to be found in these simulations. A sample movie can be found at: [http://informatics.indiana.edu/larryy/cluster\\_movie.zip](http://informatics.indiana.edu/larryy/cluster_movie.zip)



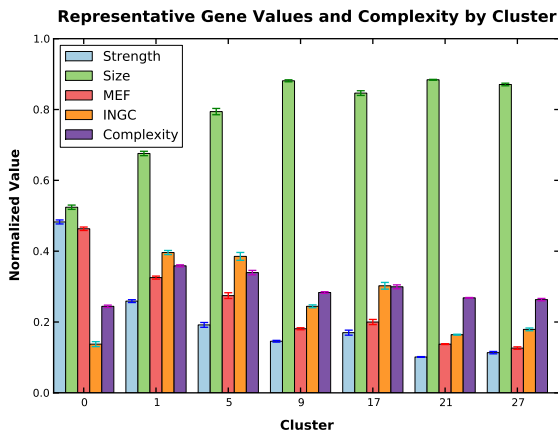


Figure 3: Means and standard error bars for the strength, size, mate-energy-fraction (MEF) and internal-neural-group-count (INGC) genes, along with neural complexity, for clusters with more than 700 agents, using  $\epsilon = 2.125$ .

### Cluster Characterization

Clustered sub-populations can be characterized by their respective cluster centroid. Figure 3 provides a set of cluster fingerprints, summarizing the raw data in Table 2 for clusters of  $\epsilon = 2.125$  (showing only the major clusters, with populations  $> 700$ ). Specific evolutionary trends can be correlated to the rise and fall of particular species. The increase in size is readily apparent, along with a general decline in mate energy fraction and strength, and a variation in the internal-neural-group count. The earlier clusters 1 and 5 explore larger neural structures, achieving higher complexity. All dominant clusters exhibit a trend towards reduced energy consumption (low mate energy fraction and strength) and increased energy capacity (large size). Cluster 17 shows an exploratory population with slightly higher internal-neural-group count and neural complexity, coupled with a reduced emphasis on energy conservation, as evidenced by an increased strength and mate energy fraction, and slightly smaller size. This exploratory population is present in the middle third of the simulation, emerging out of the dominant cluster 21, but having only limited success, and, together with cluster 16, is responsible for the bump in internal-neural-group count and complexity as previously discussed.

### Future Directions

One direction is to apply these analysis methods to simulations with simpler environments, in order to eliminate the possibility of allopatric speciation. We are also investigating methods from the evolutionary biology literature, such as “heat maps” of genetic diversity versus geological origins of parents, that might help us quantify degrees of sympatric vs allopatric speciation. An analysis of the temporal history of the fecundity and child-production rates discussed

here might help distinguish pre-zygotic and post-zygotic selection and clarify the role of reinforcement in producing assortative mating.

Alternative clustering algorithms are also of interest. Information theory-based algorithms, such as that of Gokcay and Principe (2002), which maximizes cross-entropy between clusters, look particularly attractive. Alternatively, adopting a rival-penalization method, such as the  $k^*$ -means algorithm (Cheung, 2002), may provide a better metric for cluster selection than cluster diameter. It might also be interesting to adapt the hierarchical clustering scheme of Aspinall and Gras (2010), regardless of whether we adopt their practice of allowing clusters to modulate reproductive success. Such a comparison would provide insight into whether or not varying the thresholds of QT-Clust is suggesting hierarchies of sub-populations, as hinted by Figure 2.

Any of these clustering methods, including the current one, would allow us to evaluate the effectiveness of a “miscegenation function”, which establishes a probability of reproductive success that is inversely proportional to genetic distance between potential mates, that was long ago built into Polyworld, but which has never been explored to any substantial degree.

With the existing data, a study of the geographic locality of the origin and spread of each species may yield information about environmental effects on selection and degrees of sympatric vs allopatric speciation. This may provide theoretical insights into a common real-world speciation scenario in which initial allopatric (regional) divergence is followed by sympatric divergence, as seen in Darwin’s Finches and other taxa (Huber et al., 2007). We would also like to apply these methods to simulations with clearly differentiated niches that are geographically either overlapping or isolated, to distinguish and quantify the relative effects of niche specialization vs geographic isolation.

### References

- Aspinall, A. and Gras, R. (2010). K-means clustering as a speciation mechanism within an individual-based evolving predator-prey ecosystem simulation. In An, A., Lingras, P., Petty, S., and Huang, R., editors, *Active Media Technology*, volume 6335 of *Lecture Notes in Computer Science*, pages 318–329. Springer Berlin / Heidelberg. 10.1007/978-3-642-15470-6\_33.
- Bellman, R. E. (1957). *Dynamic Programming*. Princeton University Press.
- Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. (1999). When is “nearest neighbor” meaningful? In Beeri, C. and Buneman, P., editors, *Database Theory – ICDT’99*, volume 1540 of *Lecture Notes in Computer Science*, pages 217–235. Springer Berlin / Heidelberg. 10.1007/3-540-49257-7\_15.
- Butlin, R. K. and Tregenza, T. (1997). Is speciation no accident? *Nature*, 387:551–553.
- Cheung, Y.-m. (2002).  $k^*$ -means – a generalized  $k$ -means clustering algorithm with unknown cluster number. In Yin, H.,

- Allinson, N., Freeman, R., Keane, J., and Hubbard, S., editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2002*, volume 2412 of *Lecture Notes in Computer Science*, pages 147–154. Springer Berlin / Heidelberg. 10.1007/3-540-45675-9\_48.
- Dieckmann, U. and Doebeli, M. (1999). On the origin of species by sympatric speciation. *Nature*, 400:354–357.
- Dobzhansky, T. (1937). *Genetics and the Origin of Species*. Columbia Univ. Press, New York, NY.
- Felsenstein, J. (1981). Skepticism towards Santa Rosalia, or why are there so few kinds of animals? *Evolution*, 35:124–138.
- Fretwell, S. D. (1972). *Populations in a seasonal environment*. Princeton Univ. Press, Princeton, NJ.
- Fretwell, S. D. and Lucas, H. L. (1970). On territorial behavior and other factors influencing habitat distribution in birds. *Acta Biotheoretica*, 19:16–36.
- Gokcay, E. and Principe, J. C. (2002). Information theoretic clustering. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 24:158–171.
- Hartigan, J. A. (1975). *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA.
- Heyer, L. J., Kruglyak, S., and Yooseph, S. (1999). Exploring Expression Data: Identification and Analysis of Coexpressed Genes. *Genome Research*, 9(11):1106–1115.
- Hocaoglu, C. and Sanderson, A. C. (1995). Evolutionary speciation using minimal representation size clustering. In *Evolutionary Programming '95*, pages 187–203.
- Huber, S. K., León, L. F. D., Hendry, A. P., Bermingham, E., and Podos, J. (2007). Reproductive isolation of sympatric morphs in a population of Darwin's finches. *Proceedings of the Royal Society B: Biological Sciences*, 274(1619):1709–1714.
- Kaufman, L. and Rousseeuw, P. J. (2005). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Blackwell.
- Kirkpatrick, M. and Ravigné, V. (2001). Speciation by natural and sexual selection: models and experiments. *Am Nat*, 158:S22–S35.
- Kondrashov, A. S. and Kondrashov, F. A. (1999). Interactions among quantitative traits in the course of sympatric speciation. *Nature*, 400:351–354.
- MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Cam, L. M. L. and Neyman, J., editors, *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press.
- Mallet, J. (1995). A species definition for the modern synthesis. *Trends in Ecology & Evolution*, 10(7):294 – 299.
- Mayr, E. and Provine, W. (1998). *The evolutionary synthesis: perspectives on the unification of biology*. Harvard University Press, Cambridge.
- Ortiz-Barrientos, D., Counterman, B. A., and Noor, M. A. F. (2004). The Genetics of Speciation by Reinforcement. *PLoS Biol*, 2(12):e416.
- Radovanović, M., Nanopoulos, A., and MirjanaIvanović (2010). Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning*, 11:2487–2531.
- Sætre, G.-P., Moum, T., Bureš, S., Miroslav Král, M. A., and Moreno, J. (1997). A sexually selected character displacement in flycatchers reinforces premating isolation. *Nature*, 387:589–592.
- Scharl, T. and Leisch, F. (2006). The stochastic qt-clust algorithm: evaluation of stability and variance on time-course microarray data. In Rizzi, A. and Vichi, M., editors, *Compstat 2006—Proceedings in Computational Statistics*, pages 1015–1022. Physica Verlag, Heidelberg, Germany.
- Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656.
- Silverton, J., Servaes, C., Bliss, P., and Macleod, D. (2005). Reinforcement of reproductive isolation between adjacent populations in the park grass experiment. *Heredity*, 95:198–205.
- Tononi, G., Sporns, O., and Edelman, G. (1994). A measure for brain complexity: Relating functional segregation and integration in the nervous system. *Proc. Nat. Acad. Sci.*, 91:5033–5037.
- Tregenza, T. and Butlin, R. K. (1999). Speciation without isolation. *Nature*, 400:311–312.
- Van Doorn, G. S. and Weissing, F. J. (2001). Ecological versus sexual selection models of sympatric speciation: a synthesis. *Selection*, 2:17–40.
- Wallace, A. R. (1899). *Darwinism*. Macmillan, London.
- Weissing, F. J., Edelaar, P., and van Doorn, G. S. (2011). Adaptive speciation theory: a conceptual review. *Behav Ecol Sociobiol*, 65:461–480.
- Yaeger, L. S. (1994). Computational Genetics, Physiology, Metabolism, Neural Systems, Learning, Vision, and Behavior or Polyworld: Life in a New Context. In Langton, C. G., editor, *Proceedings of the Artificial Life III Conference*, pages 263–298. Addison-Wesley, Reading, MA.
- Yaeger, L. S. (2009). How evolution guides complexity. *HFSP*, 3(5):328–339.
- Yaeger, L. S., Griffith, V., and Sporns, O. (2008). Passive and Driven Trends in the Evolution of Complexity. In Bullock, S., Noble, J., Watson, R., and Bedau, M. A., editors, *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 725–732. MIT Press, Cambridge, MA.
- Yaeger, L. S., Sporns, O., Williams, S., Shuai, X., and Dougherty, S. (2010). Evolutionary Selection of Network Structure and Function. In Fellerman, H., Dörr, M., Hanczyc, M. M., Laursen, L. L., Maurer, S., Merkle, D., Monnard, P.-A., Støy, K., and Rasmussen, S., editors, *Artificial Life XII: Proceedings of the Twelfth International Conference on the Simulation and Synthesis of Living Systems*, pages 313–320. MIT Press, Cambridge, MA.
- Zhao, Y. and Karypis, G. (2005). Data clustering in life sciences. *Molecular Biotechnology*, 31:55–80.

# Embodied reaction logic in a simulated chemical computer

Fintan Nagle<sup>1</sup> and Simon Hickinbotham<sup>2</sup>

<sup>1</sup>UCL CoMPLEX, University College London, London WC1E 6BT

<sup>2</sup>YCCSA, Department of Computer Science, University of York, Heslington, York YO10 5GH, UK

fintan.nagle.10@ucl.ac.uk      sjh@cs.york.ac.uk

## Abstract

This work uses an ALife simulation to explore the implementation of embodied reaction logic in a chemical computer. Chemical systems have potential for computation. There are properties of a logical system that are desirable in any computational system, such as the ability of the system to change state in response to some input. An issue in chemistry is that the molecules must have some physical embodiment, which must somehow represent state; state is then interpreted as the presence or absence of certain molecular configurations in the system. The design of a chemical logic gate is a means of showing that a chemical system can change state appropriately and that the information encoded in the molecules is available to be processed *as* information. This paper compares two simulated chemical computing systems: Bindworld (a simple illustrative example) and Stringmol, (a fully implemented complex DNA-inspired evolutionary computing framework). The problems and design decisions involved in creating a NOT gate in each system are compared, showing that designed computational systems require a certain complexity and flexibility to be useful to human operators. Finally we discuss general extensions to the Stringmol reaction chemistry that would simplify the process of information processing in embodied systems.

Computation is a fairly new concept to science. Although the word itself has been in use since 1447 or before, until the early 20th century it referred only to manual calculation performed by humans (this is why early machines were called “automatic computers” to distinguish them from their human counterparts). It is only since the development of the mechanical and electronic computer that the term has been applied to a process external to human thought.

Artificial Life (ALife) is a simulation of biological life on a computer. These simulations are often considered to be “embodied thought experiments” (Di Paolo et al., 2000), which test whether the essential properties of biology have been captured. Simulations of biological processes are also seen as a step towards harnessing biological processes for our own ends (Brooks, 2001). (The successful simulation will process information in a similar, but more robust, manner to our electronic computers.) It is therefore legitimate to consider how a biological simulation is capable of information processing. The simplest form of this is logic.

Models of conventional computing deliver programming languages, based on logic, that abstract the functionality away from the implementation of the logic on electrical circuitry. Programmers take this for granted when designing software using these programming languages. Similar abstractions may be needed to program computers based on other media. Generally, computation proceeds in the following context:

1. **INPUT:** An observer encodes a problem and passes it to some external system (be it electronic, neural, biological, molecular, or otherwise) via a *setup function*.
2. **COMPUTE:** The system evolves, ending in some changed state. Where the change in state has involved some notion of information processing, *computation* has occurred.
3. **OUTPUT:** The observer uses an *output function* to extract some useful information from the system and decode it into a useful response.

The problem, or series of instructions, is encoded in a different way depending on the architecture of the computer. In electronic computers it is a program in a language such as C (or, equivalently, the machine code representation of that program). The usefulness of a computer is in the **COMPUTE** stage, when the computer performs a task so that we do not need to execute it ourselves.

Systems requiring two-way data exchange (such as a search engine, which alternates between taking queries and returning results) can be seen as a series of input-compute-output operations. We do not discuss parallelism and concurrency, but restrict our discussion to an input-compute-output problem analogous to a batch processing command.

The idea of a chemical computer is appealing: somehow encode the task in a solution of *input molecules*, place them in solution with the *computational molecules*, and find the result in the set of extractable *output molecules*. Molecular computing is massively parallel - there is potential for billions of reactions to take place at the same time in a single container. For example, in DNA computing (Păun et al., 1998; Lee et al., 2004; Adleman, 1994), the setup function consists of encoding a problem in fragments of DNA. The

computation occurs *in vitro*, in parallel, and without external interference. The output function often consists of using gel electrophoresis to extract and sequence DNA of a particular weight, which we know to encode a useful response. Importantly, the information is *embodied* – it is represented as a sequence of DNA that is processed via the laws of physical chemistry.

This paper compares two embodied computers. The first is a small theoretical novel computer, Bindworld, in which simple “atoms” bind with each other to form complexes. The second is a larger, implemented novel computer, Stringmol, in which long RNA-like sequences interact according to their embedded programs. We use the comparison to highlight the challenges and tradeoffs involved in designing a novel embodied computer.

It is important to emphasise the role of the molecules-as-programs in terms of their computing power. To demonstrate the importance of reactions within a chemical computation, we briefly describe some theoretical work based on the concept of complementary binding of molecules alone. We show the shortcomings of this approach, and extend it to incorporate the concept of reaction between molecules once they have bound together before presenting our experimental implementation of a NOT gate in an artificial chemistry.

Stringmol is an *artificial chemistry* (AC) that is being developed to explore a method of evolutionary computation based on the “RNA-world” model of biology (Gilbert, 1986). This is a computational simulation in which the genome-carrier molecule is composed of the same molecular building blocks as the enzymes that are encoded therein. The system, called Stringmol (Hickinbotham et al., 2010a,b, 2009) abstracts the concept of stochastic mixing and molecular binding and reaction into a tractable model for ALife experiments. The link between computation and open-ended evolution is that both paradigms require that it is possible to generate an unbounded set of possible states. In related work (Clark et al., 2011), we have demonstrated that the sophisticated binding protocols in Stringmol are key to the diversity that the system is capable of producing. Here, we show that binding *alone* is not a convenient form for computation.

In addition to being able to carry out computation, it must also be feasible for a human programmer to initialise the system “by hand”. The idea of logic gates is an appropriate area in which to start thinking about a novel computational system, since logic is familiar and universally used in traditional computational systems. A non-standard computer which can be used with a standard computing paradigm (logic) is more approachable than one which requires an entirely new way of thinking about computation; it requires less training to use and program, and existing results and algorithms are easier to re-use. Conceptualising such a familiar idea in a new system highlights the similarities and differences between the novel system and von Neumann computing. The two sys-

tems we compare here are both formally specified (Bindworld by its reaction rules, Stringmol by its program code); these mathematical and programmatical specifications are not included here, however, as conceptual analysis of the two systems informs our main conclusions.

Logic gates have already been implemented in a real chemical system based on DNA enzymes with catalytic action (Stojanovic and Stefanovic, 2003). Simulation is an important complement to experiments with real chemistry, since the simulation can be interrogated easily and completely, and complete understanding of the molecular model is available. The issue with simulation is whether the correct properties of real chemistry have been captured in the model. The systems we compare here differ from real chemistry: Bindworld is much more simple (containing only atoms and bonds); and Stringmol uses a programming metaphor in the place of the physical properties of atoms by containing a set of operators and pointers.

Teuscher provides a good review of realisations of logic elements in chemical computers (Teuscher, 2007), focusing on how to build in reliability through redundancy in membrane-based systems. The systems we explore here, however, are different from membrane computing systems; they lack a container-based physical hierarchy.

Two important properties of a useful computational system are *preservation of state* and *change of state*. Firstly, information must be preserved in some way; the system must have some kind of memory. Secondly, information must be modified in some sensible way; a system that does not change cannot perform computation. Logic encapsulates both the idea of preservation of state (truth values are held steady) and that of change of state (output values are modified).

## What are we looking for in a non-standard computational system?

We already possess a remarkably powerful and ubiquitous computational system: the von Neumann architecture (Aspray, 1990) implemented on electronic computers. This architecture is used by virtually all electronic computing devices, from the mobile phone to the supercomputer. There is therefore little point in developing and researching a novel computational system unless it is (or has the potential to be) in some way superior to the von Neumann electronic computer. Simulations are useful for research, but often difficult to implement on traditional computers. A platform amenable to evolution is also desirable. Thus, we require a different form of embodiment to that seen in electronic computers. We seek a general computational platform that is more amenable to biological systems in general, and biological evolution in particular.

The role of a computer is to carry out algorithms for humans, or even adjust these algorithms to cope with particular problems; informally, *computers solve problems* for us.

The Church-Turing thesis (which is widely regarded and accepted as correct, but is unproved, and indeed formally unprovable (Copeland, 1996)) states that every expressible or comprehensible algorithm is computable using a Turing machine. As electronic computers are equivalent to Turing machines, no novel computer will ever allow *more* problems to be solved than an electronic computer.

One advantage of a novel computer could be that there were some problems it could solve better (either faster or more accurately). Another could be that the **INPUT** stage were easier; that the problem encoding were more understandable or easier to generate. This is very important, as much of the effort involved in computer-aided problem solving goes into formalising the problem in a way that the computer can understand. Finally, a novel computer might not allow problems to be solved faster or more easily, but might offer practical advantages like being smaller, lighter, cheaper, or more energy-efficient. A novel computer might even be able to solve fewer problems than an electronic computer, but convey significant practical advantages.

### ***In vitro* experimentation versus *in silico* simulation**

We can explore non-standard computational systems in two ways: by implementing them in the real world, or in simulation. Real-world implementation has had interesting results, such as Adleman's solution of an instance of the directed Hamiltonian path problem using DNA molecules (Adleman, 1994) and Adamatzky's reaction-diffusion logic gates in a chemical medium (Adamatzky and Costello, 2002).

Simulating non-standard computational systems has also been successful; consider Winfree's simulations of computation by interactions between self-assembling tiles (Winfree, 1998) or Adamatzky's simulations of reaction-diffusion systems (Adamatzky, 1997). Winfree simulated sets of DNA molecules with 4 "sticky ends" (ends amenable to binding with other DNA molecules) and showed that they are capable of unsupervised self-assembly, in particular patterns, into nets of DNA. This system was used to simulate a self-assembling Sierpinski triangle. Adamatzky simulated (and also constructed) logic gates whose information carriers were interacting waves of chemical reaction proceeding across a medium.

When simulating chemical computation, we need to set up an environment which approximates real chemistry. As the mechanisms of real chemistry are currently intractable (too much computation is required) and indeed not fully known, we need to set up a simplified simulation-world. It should be qualitatively similar to real chemistry, but vastly simplified to make it computationally tractable.<sup>1</sup> There are several choices we have to make:

<sup>1</sup>Unconventional computation can be a rather recursive field; we are considering the computational power of a simulation whose complexity is limited by the computational power required to simulate it!

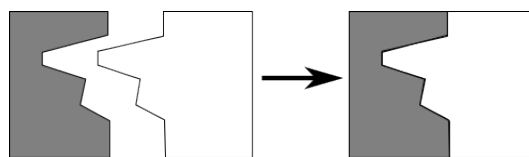


Figure 1: Two "atoms", complementary in shape, bonding together to form a complex.

- *Atoms*. What types of atoms (unsplittable entities) to include in the system.
- *Interactions*. How these atoms interact.
- *Determinism*. How large is the role of chance in the system.
- *Container*. The environment the simulation operates in; its dimensionality and boundary conditions.

### **Bindworld: a simple simulation and its drawbacks**

This section presents Bindworld, a trivial simulated chemical-analogue computational system designed around the concept of binding alone (with no explicit formalisation of reaction). This is a "thought experiment" that shows the necessity of reaction in computational chemistries.

A "program" in Bindworld consists of a population of *atoms*, so called because they are indivisible. Each atom has one or more *bindsites* of type  $k$  or  $k'$ ,  $k \in \mathbb{N}^*$ . For example, a shape could have the three bindsites of types 1, 2', 3. A bindsite of type  $k$  can only bind to one of type  $k'$ . For example, a bindsite of type 7 can *only* bind to one of type 7'. This rule reflects shape complementarity. Only bindsites on different atoms can bond; two complementary bindsites on the same atom cannot bond to each other. If a bond occurs, the two atoms in question are bonded to form a complex.

To implement a program in Bindworld, we must:

1. **INPUT**: Set up a population  $P_i$  of atoms, encoded in which are the truth values for our gate inputs. When setting up the population  $P_i$  (encoding our question) we have control of the presence or absence of atoms, and of the bindsites they possess. We can choose whether to include a certain atom in the initial population, and which bindsites to equip it with.
2. **COMPUTE**: Let the system evolve by forming all possible bonds between atoms.
3. **OUTPUT**: Interrogate the final population  $P_f$ , and infer the state of the system from the pattern of bonds found between atoms.

Bindworld can either be deterministic (if two atoms can bind at a particular moment, they certainly will) or non-deterministic (if two atoms can bind, they may either bind or remain unbound).

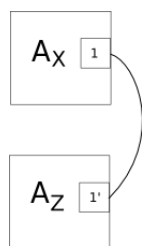


Figure 2: A molecular NOT gate; atom  $A_X$  (top) has bound to atom  $A_Z$  (bottom) via complementary bindsites of types 1 and 1'. Here the case where  $X = \text{true}$  is illustrated; the atom  $A_X$  is present. The case  $X = \text{false}$ , this atom would be absent and  $A_X$  would be bound to nothing.

One of the implications of having bonding with no reaction is that when reading the final population  $P_f$ , it is useless to read either the presence/absence of atoms or which bindsites they possess, as this information will be the same as in the initial population. The only way in which the system can change is by bonding atoms to other atoms and forming complexes. We must therefore extract the output value of our logic gate from information about which atoms are bound to which.

### Making logic gates in Bindworld

Suppose we have access to two binary variables  $X$  and  $Z$  and two atoms  $A_X$  and  $A_Z$  with complementary bindsites 1 and 1'. These atoms are notated  $A_{X,1}$  and  $A_{Z,1'}$  and will bond to each other as their bindsites are complementary. We then define the input rule and the output rule formally as:

$$\begin{aligned} \text{Input: } P_i &= \begin{cases} \{A_X, A_Z\} & \text{if } X = T \\ \{A_Z\} & \text{if } X = F \end{cases} \\ \text{Output: } Z &= \begin{cases} F & \text{if } \text{bound}(A_Z)^{P_f} \\ T & \text{if } \neg \text{bound}(A_Z)^{P_f} \end{cases} \end{aligned} \quad (1)$$

We set the output variable  $Z$  to true if atom  $A_Z$  is bound to anything in  $P_f$ , shown by the helper function  $\text{bound}(A_Z)^{P_f}$ . This implements the relation  $Z = \neg X$  in Bindworld; the chemistry of the system behaves differently depending on how we set up the initial population.

As shown, it is simple to engineer one gate on its own. It is also simple to run a set of gates in parallel by simply using sets of bindsites which do not interact with each other. It is when we come to link gates together by connecting the output of one to the input of the next (as we must to run any meaningful computation) that problems occur.

One way of doing this is with the NOT gate is by simply inserting another atom  $A_I$  with a bindsite of type 1. If we do this *after* the initial gate has run, the results will be sensible and  $\neg \text{bound}(A_Z)^{P_f}$  will reflect  $\neg \neg X$ . However, this means the computer's execution would have to be paused after the

operation of every gate, which would make the computer's operation intolerably slow, as such chemical manipulations are very slow. Furthermore, they would need to be controlled either by hand or by an electronic computer; using a traditional computer to facilitate the operation of a novel one (when it could just evaluate the gates electronically) is a waste of resources. We want to be able to program the computer and leave it to run unsupervised.

The process is even more complex when dealing with 2-input gates such as AND or NOR, which raises another disadvantage of Bindworld: the activity of programming it (setting up the atoms and the initial population) is very hard. It is conceptually very nonintuitive and difficult to reason about. This violates another goal in the design of novel computers, that they should be easy to control and program. There may be an arcane, complex way of setting up unsupervised chained gates in Bindworld, involving helper bindsites and ancillary atoms, but it escapes us.

To sum up, Bindworld has several down sides. Firstly, programming it is extremely nonintuitive. Secondly, it seems even simple one-input-gate chaining is very hard unless the population can be adjusted after the evaluation of every gate, a restriction which would cripple the system's power. Thirdly, although we can define the joining of two atoms as a "reaction," Bindworld has no *explicit* concept of reaction, which means it does not integrate very well with our mental schemas of computation and chemistry. The next section describes an artificial chemistry built around this concept.

### Computation with the Stringmol artificial chemistry

We give here a brief overview of our molecular system, which is described fully in Hickinbotham et al. (2010a, 2009). A summary of the chemistry is presented below, followed by a description and discussion of molecular structure.

In order to express our observations in a computational system, we identify three major domains. The first defines the underlying *physico-chemical properties* of the atomic components. The second is the coding of the proteins in genes - the *sequence of codes*. The third is the *embodiment* of both the physico-chemical properties and the sequence of codes in the physical world. The physico-chemical properties of the system are immutable, but they specify a space of possible realisations that is immeasurably vast. Genetic systems are specified by the sequence of codes, but importantly, the sequence is *embodied* within the system, thus allowing the enzymes that the sequence encodes to act on the embodiment of the sequence itself and thus modify it. This has places the sequence management apparatus under control of the sequence itself, eventually exploiting the available possibilities that the physico-chemical properties endow upon the embodied system. This phenomenon is the basis of bi-

ological evolutionary systems, where the embodiment of a genetic code in a carrier molecule allows the encoded proteins to “curate” the genome. Initial experiments by Hickinbotham et al. (2010b, 2009) into implementing such a system have resulted in the Stringmol chemistry.

## Molecular representation

In Stringmol, the molecular analogues are composed of sequences of token symbols (single letters or symbols such as ‘\$’ or ‘>’) which represent both the structure of the molecule and a series of programmatic instructions. Molecules bind at loci along sequences if there is a match between the sequences at that point. Importantly, the match is inexact, and is modelled as a probability of a bind occurring. The basis of the soft alignment scoring function is based on the scoring method of Smith and Waterman (1981).

Once bound, the two molecules have the potential (by following the programs specified by their strings of instructions) both to create new molecules and to change their composition, thus forming new molecules. This is the reaction component of the system. The sequence is treated like a program, commencing at the beginning of whichever aligned subsequence is furthest from the beginning of its string. There are 7 functional symbols, shown as non-alphabetical characters ‘\$’, ‘>’, ‘^’, ‘?’, ‘=’, ‘%’, and ‘}’. Stringmol uses functional symbols to specify the manipulation of a set of pointers which indicate positions on the molecular strings, and the symbols that the pointers index. During a reaction, alignments are used to specify program flow, commonly acting as place markers and analogues of “goto” statements.

Note that in Stringmol, binding and reaction are completely chronologically and conceptually separate. Once a bind is effected, it remains in place for the duration of the reaction. Another bind cannot interrupt a reaction; a third Stringmol cannot bind to a reaction in progress.

## System Architecture

A Stringmol simulation can be considered as a set of reacting molecules whose movements inside a container are governed by a stochastic mixing function. All molecules are subject to *decay* (spontaneous destruction), which places a requirement upon the system to act in order to maintain itself in the face of entropy. Should molecules come sufficiently close to one another, then they can *bind* and subsequent to binding occurring, *react*. The system has a discrete-time clock. At each time step, all the molecules in the system are processed. Actions only occur if energy is available. Energy is consumed via binding and executing each instruction in a reaction; these two events each have an energy cost. The likelihood of binding and the nature of the reaction is encoded in the string of each molecule in the encounter. At one particular time step, we specify that 25 energy units are available. The selection of which events consume the energy

is stochastic. The balance between energy availability and the decay rate of the molecule maintains a steady population of molecules. We currently specify that only two molecules can ever participate in a single reaction, and that raw materials for the assembly of new molecules are available in saturation.

## Strategy for implementing molecular logic

Our implementation of logic within this artificial chemical system allows us to demonstrate the ability of the system to change state. There are two points arising from this. Firstly, the implementation of the processing is not straightforward, since the reaction-language was not tailored to logic. Secondly, following from the first point, there naturally arises within the system the possibility of evolving the system to deliver fuzzier logical analysis from the initial bootstrap, via a built-in ability of the system to evolve.

We require that the chemical system acts to maintain a population of molecules in an environment where no molecule can persist indefinitely. We thus base our system on a molecular species that we call a *replicase*, R. This molecule R has embodied properties coded into its sequence that allows it to bind to copies of itself and create further copies.

Before an input data molecule D enters the system, the R molecules simply maintain a stable population. The input does not persist in the system, so in order for the system to generate a response that does persist, the input signal must induce two changes in R. In our implementation, R is ‘primed’ to implement changes to its own sequence when D binds to a specific region, to introduce a signal generating molecule S that not only acts as a replicase, but also generates an output molecule O.

Experiments showed (Hickinbotham et al., 2010b) that changes in the binding site of the replicase allowed new species of replicase to be preferentially copied, and thus drive other replicase molecules to extinction. We exploit this phenomena in the design of our state-change when an input data molecule appears – it changes the sequence of the replicase molecule R it interacts with to always *be copied* rather than *act as the copier*. This means that the original replicase R is swept out of the system, and a new replicase species S takes its place. S not only self-maintains, but also produces *output molecules*.

## Designing the molecular species

The sequences of logical Stringmol data input molecules D must perform two tasks. Firstly, they must bind to the replicase molecule, and secondly, they must encode the logical state of the input. For a “true” signal, we specify the sequences INPUTTTTTT and INPUTFFFFF for false. There are two regions to this molecule. The sequence INPUT forms the bind site (shown in yellow in figure 3II, where



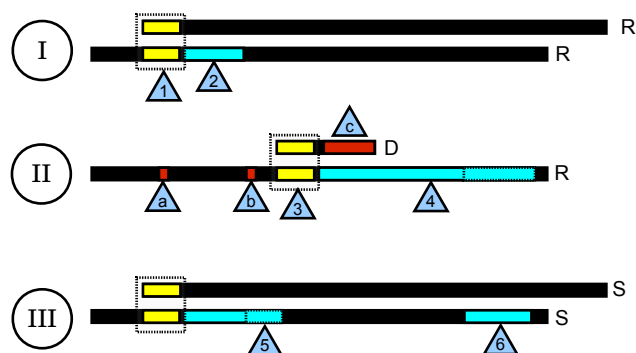


Figure 3: The replicase NOT gate molecules. Yellow regions are bind sites, blue regions are program sequences, red regions are read/write areas external to the program regions. Dotted lines show binds. Black regions are parts of the sequence that are not used in the reaction. Particular regions of molecules are referenced by numbers and letters in triangles. There are three reactions **I**, **II** and **III**: **I**: The replicase **R** will copy anything that binds at position 1, using the program encoded at region 2. **R** can bind to other **R** molecules. **II**: The data molecule **D** binds to **R** at region 3. The program at 4 is executed, which changes sites **a** and **b** and uses the logic encoding at **c** to write a specification of the output molecule. This changes the sequence of molecule **R** to create a new molecule species **S**. **III**: The signal-producing molecule **S**. This molecule has the dual functionality of copying the molecule it binds to, and moving program flow from region 5 to region 6, where the output molecule is expressed.

as the sequences **TTTTT** or **FFFFF** encode the logical state (shown in red).

The central challenge in this experiment was designing the replicase molecule **R** to bind and process the input molecule. The string encoding the functionality for this molecule was 243 symbols long. The reactions that the sequence encodes are shown in figure 3. There are three reactions to consider. Reaction **I** is the replication function, encoded on region 2 the top row of the figure. We refer the reader to Hickinbotham et al. (2009) for a discussion of the replicase functionality encoded in this region. Reaction **II** occurs when **D** binds to **R** at region 3. The processing of the input molecule is encoded in region 4, and proceeds as follows:

1. **check-input**: Inexact alignments can form a bind with low but significant probability. It is therefore important to check the validity of molecules which bind at region 3. The sequence `?VACH` carries out this check, and terminates the reaction if the condition is not met.
2. **Mod-replicase**: This sequence carries out the modification of the replicase bind site at region **a**, and deletes the terminate symbol `}` at region **b**. This change means

- that region 5 will be executed in reaction **III** to initiate production of the output molecule when **S-S** binds occur.
3. **Check-boolean**: Switches program flow to create a template for an output molecule that embodies “true” or “false” in the system.
4. **Set-output-false** and **Set-output-true**: These sequences position the read pointer over the sequence encoding “true” or “false” respectively depending on the output of a NOT operation on the logical state of **D**.
5. **Make-output-message** writes the output of the NOT operation into the template sequence for the output molecule.
6. **Express-output-message** creates a new output molecule. Note this sequence is also shown in reaction **III** as region 6. The program executed by the **S** jumps to this region from region 5.

The output molecules are **ODTWKZFFFFF** for false and **ODTWKZTTTTT** for true. Note that we had originally coded this molecule using the sequence **OUTPUT**, but the last three letters of this sequence formed a partial match with the bind site for **INPUT**.

It is clear the mechanism for latching the system is more complex than in an electronic logic gate. This is a consequence of the fact that *everything* in the system is subject to decay, so in order to preserve the output, it must be repeatedly created by **S**. However, this does provide the facility for new configurations to arise by allowing mutation to occur in the system as in (Hickinbotham et al., 2009).

## Experimental trial

To demonstrate the effectiveness of the molecular specification, we ran 1,000 trials each of input conditions with true, false and **NULL** configurations. A previously implemented C++ incarnation of Stringmol was used to run the trials, all of which successfully produced the output signal molecule, maintaining the population indefinitely. Examples of processing a true and false input signal are shown in figure 4 (the null configuration is simply a constant population of the seed replicase). In both of these examples, the molecular population dynamics are similar. The Input signal binds to the Start Replicase, which executes the self-modifying code. This produces the “signal replicase”. We see the population of Start Replicase drop off more quickly than the input molecule, since it is subject to modification into the Signal replicase *and* decay, whereas the input molecule is only subject to decay. The bump in the population of the Signal replicase is due to an energy glut, since the input population is too small to consume available energy. Finally, we see the emergence of a steady state population of two molecular species: the Signal replicase, and the output molecule encoded with the appropriate boolean NOT response.

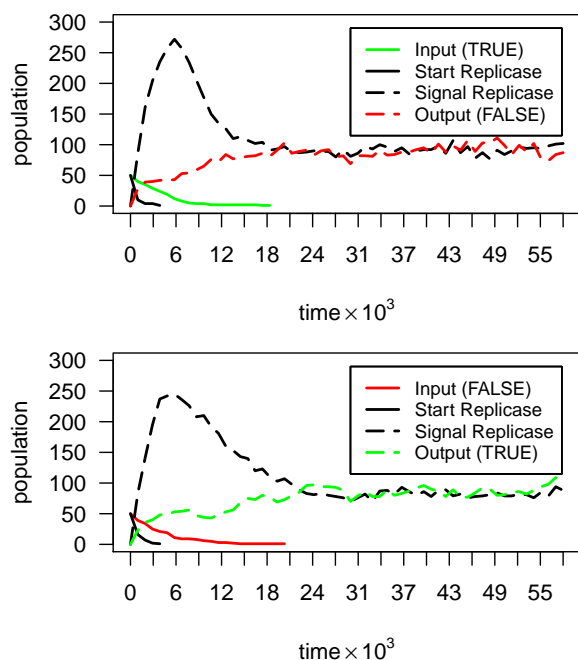


Figure 4: Reaction logic processing of an input molecule through a NOT replicase. **Top:** “true”, encoded as INPUTTTTTT (green line), when combined with the “start replicase” (black line), creates a population of output “false” molecules encoded as ODTWKZFFFFF (red dashed line), along with a Replicase-plus-signal enzyme (black dotted line). **Bottom:** Same as the above, but with “false”, encoded as INPUTFFFFF (red line), and a population of output “true” molecules encoded as ODTWKZTTTTT (green dashed line).

## Comparison and conclusion

We have shown theoretically and experimentally how an embodied reaction process enhances the computational power of chemical systems. The main demonstrator for this has been the design of a simple NOT gate. The differences between Bindworld and Stringmol mirror several important considerations in the design and engineering of novel computational systems.

A major difference is that Stringmol is more complex than Bindworld; it has more atoms, more complex combination rules, and a clear, hardwired border between binding and reaction. Informally, Bindworld is a simpler *world* than Stringmol. This means that, to express complex ideas like logic gates in the terms of Bindworld, we have to do more work to reduce them to its simple terms. Stringmol (like a high-level programming language) has more useful abstractions that we can incorporate when designing “code” that runs in Stringmol. Its programmatic instructions encode the reaction potential of each entity. Stringmol is in closer cor-

respondence with our mental schemas of the problems we wish to solve. As ease of programming is a very important computing property, this consideration is important. Stringmol vs. Bindworld is an example of how making a system more complex can make it easier to use.

Stringmol is also nondeterministic; this can be an advantage because it allows the same starting state to cause different behaviours, which may occur at different rates and can aggregate over time into more complex behaviours. Von Neumann computation relies on the permanent assumption of determinism, but this is not necessary (many biological computers, like the brain, do not require it) and means we have to work harder to implement nondeterminism (as in random number generators). It also leads us into a procedural, deterministic, local mental schema of computing, which is not something we want to be crippled by when designing distributed, concurrent or nondeterministic systems.

With artificial chemistries like Stringmol, there naturally arises the possibility of evolving the system to deliver fuzzier logical analysis from the initial bootstrap, via a built-in ability of the system to evolve. We plan to explore this avenue in future work, with a view to delivering a system capable of evolving solutions within the embodied chemistry.

In our experimental work, we have taken pains to develop a solution that required no changes to the artificial chemistry that was used in (Hickinbotham et al., 2010b) for applications in evolution. This is important, since biological evolution exploits the embodiment of the genome in much the same way as the embodied reactions we explore here. We had to overcome some difficulties with the functional codes in Stringmol, and also some difficulties with similarities in binding sequences that led to mis-alignments. These indicate that the Stringmol system would have more expressive power in both evolutionary and computational experiments were these issues addressed.

Rather like Newspeak in Orwell’s 1984, the expression of certain things in Stringmol is difficult if not well-nigh impossible. Intuitively, the process of simulating the physical copying of a sequence of letters seems to require more information processing than a single logic gate. However the process of copying information is not the same as actually processing the information itself. Thus in Stringmol it is easy to copy strings, but it is very difficult to express a straightforward logic gate because there are not the straightforward expressions available to do so.

We note that simple changes to the Stringmol language that would emphasise the concept of molecular embodiment of information would make the logic gate easier to implement. These are: • **Cut and paste** of strings, rather than copy and paste. Currently, we have to laboriously copy each symbol in a string to do some information processing. But some of these operations do not really require copies to be made. We could just as easily use what is being copied.

Thought of as an embodied system, the advantages of this are clear. Cut and paste in Stringmol would mean that subsequences could be excised and spliced into other sequences by manipulation of pointers. • **Regulation** of some behaviour could occur if repressors could be made possible. In our NOT gate, the input signal re-programmed the seed replicase so that it made an output signal. This programming would not have been necessary in a system where regulation of expression was feasible. • **Energy-dependent behaviour:** Our processing system is subject to energy constraints. If we could switch when energy was low, we could change behaviour to correct it. This would allow regulation of energy to occur and give rise to selection at the molecular level that is not currently possible • **More steps to copy:** If we could dismantle the '=' operator, we'd be able to do more sophisticated construction of signals. As it is, we have to string ===== together to copy short sequences, that is not obviously evolvable without six corresponding mutations. If we could use the Nellis-Stepney system, we could increment the write pointer without incrementing the read pointer, and thus have a copy of a symbol many times. • **Increment direction:** If we could switch this, it would be possible to write/evolve more compact programs. • **Pointer referencing:** If we could move *any* pointer to any other pointer, rather than the limited set currently implemented, we could more easily implement certain information processing behaviours. probable alignments.

It is interesting to note that many of these extensions could be applied to other string-based ALife systems, such as Tierra (Ray, 1991), Avida (Johnson and Wilke, 2004) and Typogenetics (Gwak and Wee, 2007). These systems were also designed to carry out the task of replication, and they are known to have limitations in terms of evolutionary openness. Our studies here indicate that there is potential to extend the instruction sets in these models to allow richer information processing, which may lead to richer behaviours.

Stringmol and Bindworld are doubtless far from any useful chemical computer, being after all only simulations. However, they allow us to explore the properties of chemical and bio-inspired long molecule computing, a strategy which we hope will eventually allow us to design a useful biological computer.

## Acknowledgements

The authors thank Susan Stepney, Peter Young, Tim Clarke, Edward Clark and Adam Nellis for comments and suggestions during the preparation of this manuscript. Simon Hickinbotham is funded by the Plazmid project, EPSRC grant EP/F031033/1.

## References

- A. Adamatzky and B. D. L. Costello. Experimental logical gates in a reaction-diffusion medium: The XOR gate and beyond. *Physical Review E*, 66(4):046112.1–046112.6, 2002.
- A. I. Adamatzky. Universal computation in excitable media: the 2+ medium. *Advanced materials for optics and electronics*, 7(5):263–272, 1997. ISSN 1099-0712.
- L. M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021–1024, 1994.
- W. Aspray. John von neumann and the origins of modern computing. *Technology and Culture*, 33(1), 1990.
- R. A. Brooks. Steps towards living machines. In *ER 2001*, pages 72–93. Springer-Verlag, 2001.
- E. Clark, S. J. Hickinbotham, S. Stepney, T. Clarke, A. Nellis, M. Pay, and J. P. Young. Degeneracy enriches artificial chemistry binding systems. In *ECAL 2011*. MIT Press, 2011.
- B. J. Copeland. The Church-Turing thesis. *Stanford encyclopedia of philosophy*, pages 1095–5054, 1996.
- E. A. Di Paolo, J. Noble, and S. Bullock. Simulation models as opaque thought experiments. In *Alife VII*, pages 497–506. MIT Press, 2000.
- W. Gilbert. Origin of life: The RNA world. *Nature*, 319(6055): 618, February 1986. doi: 10.1038/319618a0.
- C. Gwak and K. Wee. Construction of hypercycles in typogenetics with evolutionary algorithms. In *ECAL 2007*, pages 1060–1068. Springer-Verlag, 2007.
- S. J. Hickinbotham, E. Clark, S. Stepney, T. Clarke, A. Nellis, M. Pay, and J. P. Young. Molecular microprograms. In *ECAL 2009*, LNCS, pages 291–298. Springer, 2009.
- S. J. Hickinbotham, E. Clark, S. Stepney, T. Clarke, A. Nellis, M. Pay, and J. P. Young. Specification of the stringmol chemical programming language version 0.1. Technical Report YCS-2010-458, Univ. of York, 2010a.
- S. J. Hickinbotham, A. Faulconbridge, and A. Nellis. The blind watchmaker's workshop: Three artificial chemistries in the context of eigen's paradox. In *Alife XII*. MIT Press, 2010b.
- T. J. Johnson and C. O. Wilke. Evolution of resource competition between mutually dependent digital organisms. *Artif. Life*, 10:145–156, April 2004.
- J. Y. Lee, S. Y. Shin, T. H. Park, and B. T. Zhang. Solving travelling salesman problems with DNA molecules encoding numerical values. *BioSystems*, 78(1-3):39–47, 2004.
- G. Păun, G. Rozenberg, and A. Salomaa. *DNA computing: new computing paradigms*. Springer Verlag, 1998.
- T. S. Ray. An approach to the synthesis of life. In *Alife II*, pages 371–408. Addison-Wesley, 1991.
- T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–197, 1981.
- M. N. Stojanovic and D. Stefanovic. A deoxyribozyme-based molecular automaton. *Nature Biotechnology*, 21(9):1069–1074, 2003.
- C. Teuscher. Exploring logic artificial chemistries: An illogical attempt? *CoRR*, abs/cs/0703149, 2007.
- E. Winfree. Simulations of computing by self-assembly. *DIMACS: DNA-Based Computers*, pages 213–242, 1998.

# Virtual Fluid Environment on Behavior Ability for Artificial Creature

Keita Nakamura<sup>1</sup>, Ikuo Suzuki<sup>1</sup>, Masahito Yamamoto<sup>1</sup> and Masashi Furukawa<sup>1</sup>

<sup>1</sup>Graduate School of Information and Science Technology, Hokkaido University, Japan  
{ poco, ikuo, masahito, mack } @complex.ist.hokudai.ac.jp

## Abstract

An environment plays an important role in behaviors acquisition for artificial creatures. Thus, the environment must obey the physical laws. In this paper, it is examined how the behavior differences appear when the artificial creature autonomously behaves in some fluid environments. We construct the approximate virtual fluid environment with low computing costs to simulate the behavior acquisition for artificial creatures. Also we propose a simulation method for artificial creatures in consideration of effects from the virtual fluid environment based on physics modeling. As a result of simulation, we verify that it is possible for the creature to acquire adaptive behaviors in different environments. After evolution, the creature behaves autonomously by leveraging effectively fluid forces in each virtual environment.

## Introduction

Many computer simulations have been performed for studying acquiring behaviors, evolution, and learning methodologies on virtual artificial life creatures in the field of Artificial Life (ALife) and evolving robotics. Artificial fish swims automatically by learning its behavior controller (Tu and Terzopoulos, 1994). This study makes it easy to create fish animation. A flock simulation approach is developed based on a distributed behavioral model without setting the orbit of each bird (Reynolds, 1987). This approach makes it easy to create flock animation. The virtual creature is able to acquire its morphology and behavior by an evolutionary methodology based on the creature's competition (Sims, 1994a)(Sims, 1994b). Many studies for behavior acquisition are based on Sims' studies. Sims' model is applied to the virtual catapult creatures to evolve (Chaumont et al., 2007). This creature could throw its parts of body as far as possible. The relation for co-evolution of virtual creatures is observed by fighting each other in Sims' virtual environment (Miconi, 2008). In these days, there are many simulations for artificial creature using the physical calculating engine. It enables these creatures to obey physics law easily. "Snake-Like Robot" acquires adaptive locomotion on the ground using it (Tanev et al., 2005). This model is robust for obstacles. In these studies, the experimental environment is set

as an ideal environment in a computer simulation space because they considered that the methodology of evolving and learning behavior in an ideal environment is more important than acquisition of the similar behavior in realistic environment. Therefore, the influenced force from the fluid environments to the artificial creature is not precisely analyzed. Instead, the implemented force adopts the simple calculation methods for reducing the computing time. On the other hand, in a field of numerical fluid dynamics, many fluid simulations are based on a finite element method and a particle method. The moving particle semi-implicit method makes it easy to create animation on the water surface (Koshizuka et al., 1998). A virtual anomalocaris model swims in the virtual two-dimensional water environment using the particle method (Usami, 2007). And an artificial creature behaves based on a rule method considering the fluid effect (Lentine et al., 2010). The finite element method and the particle method give accurate results. However, they consume much computational time. Therefore, it is unsuitable for a real-time simulation to acquire appropriate behaviors in the virtual fluid environment. However, we consider that the virtual environment needs to acquire a more natural policy of adaptive behaviors.

In this paper, it is examined how the behavior differences appear when the artificial creature autonomously behaves in the different fluid environments. At first, we construct an approximate virtual fluid environment which enable us to do the behavior acquisition simulation with low computing costs. This environment is constructed by setting physics parameters such as the fluid density and drag coefficients. And we propose a simulation method for the artificial creature in consideration of the fluid environment. The artificial creature imitating a flat fish is modeled by connecting rigid bodies. This creature can behave by moving its bodies. In order to control bodies and learn the behaviors, an artificial neural network (ANN) is implemented with the creature. Genetic algorithm (GA) is applied to the ANN by its evolution. We experiment to examine how the artificial creature can acquire adaptive behaviors in some fluid environments. As a result of simulation, we verify that it is possible to ac-

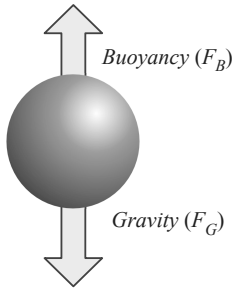


Figure 1: Modeling buoyancy

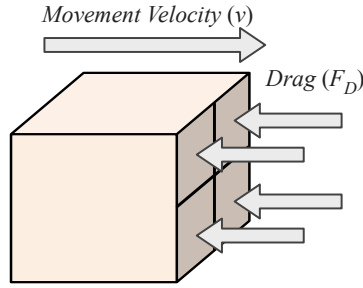


Figure 2: Modeling drag

quire an adaptive behavior for the artificial creature in virtual fluid environments. And we analyze the acquired behaviors by examining a relation between fluid environments and acquired behaviors.

### Construction of the Virtual Fluid Environment

We assume that the buoyancy and drag act as the forces that virtual rigid objects (sphere, rectangular parallelepipeds) receive from the fluid effect. We construct a virtual fluid environment by modeling two forces acting on the object in the fluid environment. These two forces compare to the buoyancy and drag respectively. The simulation is performed by calculating objects' movement, which obeys a physics law, resulting in an animation. We employ "PhysX (offered by the NVIDIA Corporation)" as a physical calculating engine. PhysX is applied to calculate a basic physical operation, for example, a gravity, a friction force, and collisions among virtual objects. In the virtual fluid environment the acceleration of the gravity  $g$  is  $9.807[m/s^2]$ . We construct the fluid environments by changing the parameter of fluid density  $\rho$ .

#### Modeling Buoyancy

Based on Archimedes' principle, we model the buoyancy as a force whose strength  $F_B$  equals to the weight of the fluid volume which an object occupied in the fluid. This force acts on the center of the mass in the opposite direction of gravity (Fig.1). The strength of the buoyancy in the fluid environment,  $F_B$  [N], is given by equation 1,

$$F_B = \rho V g \quad (1)$$

where  $\rho[kg/m^3]$  is the density of the fluid,  $V[m^3]$  is the volume of the object, and  $g[m/s^2]$  is the acceleration of the gravity.

#### Modeling Drag

Based on fluid dynamics, we model the drag as uniformly distributed forces to the surface of the object in the opposite moving direction (Fig.2). In the field of fluid dynamics,

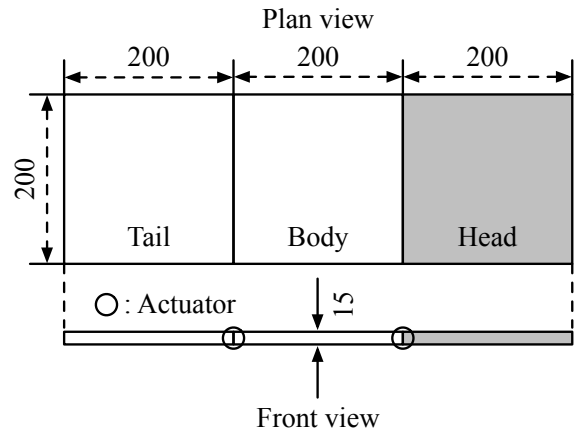


Figure 3: Artificial flat fish model

$F_D[N]$  is given by equation 2,

$$F_D = C_D \frac{1}{2} \rho v^2 S \quad (2)$$

by using dynamic pressure of a flow  $\frac{1}{2} \rho v^2 [kg/(m \cdot s^2)]$  derived analytically as the strength of the drag in the fluid, where  $C_D$  is a scalar quantity called the drag coefficient, and  $S[m^2]$  is the reference area of the object. The drag coefficient depends on the shape of the object. In this study, the drag coefficient of a rectangular parallelepiped is 1.50. The reference area of the object is the projection area of the object to the plane which is perpendicular to the flow.

An artificial creature can generate a propulsion force by moving its bodies because the modeled drag force is added to its bodies when this creature moves its bodies.

### Experiment for Acquisition Behavior in the Different Fluid Environment

We examine how the differences appear when an artificial creature autonomously behaves in some fluid environments. It is assumed that the model must move forward as efficiently as possible. Evolutionary computing is adopted to acquire the adaptive behavior.

#### Artificial Flat Fish Model

We model the artificial creature by connecting rigid bodies with actuators. The modeled artificial creature imitates a flat fish, which can behave by controlling its bodies. After evaluation of this model by evolutionary computing in fluid environments, this creature behaves effectively by using leverage fluid forces in each virtual environment. Figure 3 shows an artificial flat fish model. The fish model consists of three rectangular parallelepiped with same sizes. This model has two actuators with one degree of freedom (Fig.4). The density of the model is as same as that of fluid.

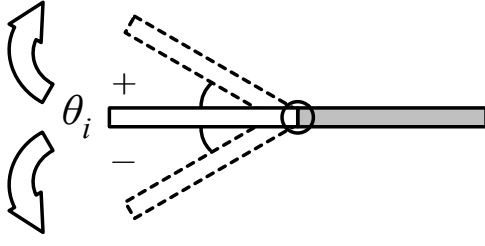


Figure 4: Model's actuator

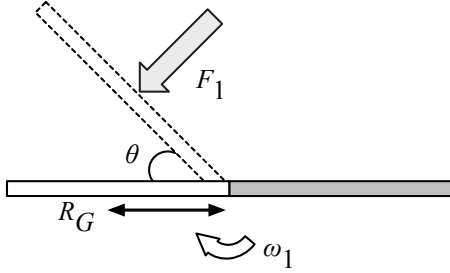


Figure 5: Modeling behavior (to bend model's body)

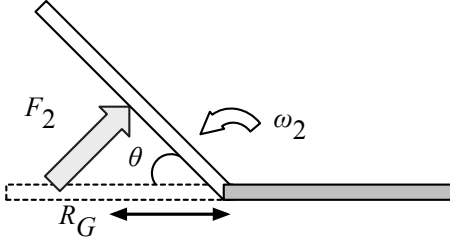


Figure 6: Modeling behavior (to unbend model's body)

### Modeling Behavior for Flat Fish Model

We focus on one actuator, when the model bend upward its body, the center of gravity of its body received drag force  $F_1$  (Fig.5). The strength of the drag force  $F_1$  is given by 3,

$$F_1 = \frac{1}{2} C_D \rho S v_1^2 \quad (3)$$

where  $\theta$  is the bend angle,  $r_G$  is the distance between the actuator and the center of gravity of its body,  $\omega_1$  is the angular velocity of the actuator and  $v_1^2$  it the speed of the body.  $v_1$  is given by 4,

$$v_1 = r_G \frac{d\theta}{dt} = r_G \omega_1 \quad (4)$$

Therefore,  $F_1$  is expressed by 5

$$\begin{aligned} F_1 &= \frac{1}{2} C_D \rho S (r_G \omega_1)^2 \\ &= k \omega_1^2 \left( k = \frac{1}{2} C_D \rho S r_G^2 \right) \end{aligned} \quad (5)$$

In the same way, when the model unbends its body, the center of gravity of its body receives the drag force  $F_2$  (Fig.6), The strength of the drag force  $F_2$  is given by 6,

$$F_2 = k \omega_1^2 \quad (6)$$

where  $\omega_2$  is the angular velocity of the actuator.

In order to move the model forward, the equation 7 is satisfied.

$$\int_{\theta}^0 k \omega_2^2 \sin \theta d\theta - \int_0^{\theta} k \omega_1^2 \sin \theta d\theta > 0 \quad (7)$$

By solving 7, the following relation

$$k(\omega_2^2 - \omega_1^2)(1 + \cos \theta) > 0 \quad (8)$$

$k$ ,  $1 + \cos \theta$  are the positive value,

$$\omega_2 - \omega_1 > 0 \quad (9)$$

This equation 9 means that the model moves forward by the speed of unbending the body faster than that of bending the body. In the same reason, when the model bends its body downward, the model moves forward by the speed of unbending the body faster than that of bending the body, too.

Therefore, the model moves forward as efficiently as possible by the speed of unbending the body, which is faster than that of bending the body on each actuator.

### Control Method for Flat Fish

An artificial neural network (ANN) is introduced to move flat fish model's actuators autonomously depending on information given by its sensor and actuators. Actuators are controlled by outputs of the three-layer feed-forward ANN. Table 1 shows the input and output parameters of the ANN. A transfer function for the ANN  $f(x)$  is formalized by combining two sigmoid functions, given by equation 10.

$$f(x) = \frac{1}{1 + e^{(-\frac{x}{\alpha} - \beta)}} + \frac{1}{1 + e^{(-\frac{x}{\alpha} + \beta)}} - 1 \quad (10)$$

Figure 7 shows an example of the transfer function ( $\alpha = 0.1$ ,  $\beta = 5.0$ ). This function enables the ANN to output the zero value. The number of neurons in the hidden layer is the same as the number of neurons in the input layer. Synaptic weights of the ANN are initialized by a real random number at first. The model enables itself to behave more effectively by optimizing synaptic weights of the ANN and the gain of the transfer function.

### Experimented Condition

We experiment to examine how the differences appear when the artificial creature autonomously behaves in some fluid environments. The flat fish model must move forward as efficiently as possible within a definite period of time (Fig.8).

Table 1: Setting of input and output parameters of ANN

Input	Relative angle of actuator $i$ between rectangular parallelepiped in each time ( $\theta_i$ )
	Relative angular velocity of actuator $i$ between rectangular parallelepiped in each time ( $\omega_i$ )
Output	Object angle of actuator $i$ between rectangular parallelepiped in each time ( $A_i$ )

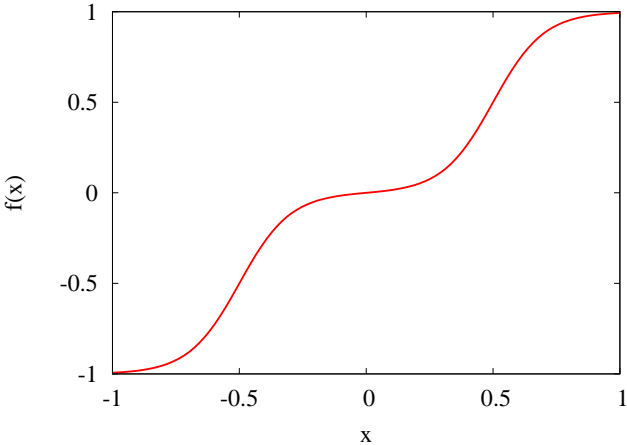


Figure 7: Transfer function for ANN ( $\alpha = 0.1, \beta = 5.0$ )

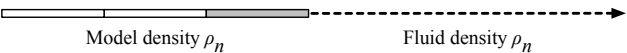


Figure 8: Initialize state of a experiment (Front view)

We artificially prepare six fluid environments for experiments. Table 2 shows the density  $\rho_n$  used for each fluid environment. The GA optimizes the synaptic weights of the ANN and the gain of transfer function by applying the RCGA. Table 3 shows ANN and RCGA conditions for this experiment. An evaluated value for the RCGA as a fitness function is set so that the creature moves forward as possible as it can. This evaluated value  $F_{eval}$  is given by 11.

$$F_{eval} = \sum_{t=0}^{Step} x(t) \quad (11)$$

where  $Step$  is the number of step used for the simulation at each generation,  $x(t)$  is a distance from a start position at each simulation step  $t$ .

Result and Discussion

We upload the movies to URL (<http://autonomous.complex.eng.hokudai.ac.jp/researches/physics-modeling/movies/nakamura>) that flat

Table 2: Experimental condition

ANN	The number of the neuron in the input layer	5
	The number of the neuron in the hidden layer	5
	The number of the neuron in the output layer	2
	The range of an object angle	$[-30^\circ, 30^\circ]$
GA	Genotype	$Weight_{ij}, \alpha, \beta$
	Phenotype	$F_{eval}$
	Population	30
	1 Step	$1/60[s]$
	Simulation step	300
	Generation	250
	Crossover Probability	0.05
	Mutation Probability	0.85
	Trial times	30

Table 3: Density of each fluid environment

$\rho_1$	1.20[kg/m <sup>3</sup> ] (Air environment)
$\rho_2$	200.0[kg/m <sup>3</sup> ]
$\rho_3$	400.0[kg/m <sup>3</sup> ]
$\rho_4$	600.0[kg/m <sup>3</sup> ]
$\rho_5$	800.0[kg/m <sup>3</sup> ]
$\rho_6$	998.20[kg/m <sup>3</sup> ] (Water environment)

creature acquires adaptive behaviors. Figure 9 shows a diagram which draws the position of best individual at each simulation time in each environment. The angle between rigid bodies on the best individual in each environment is shown in Fig.10-15. From these results, model's bodies oscillate periodically, and the angle between its bodies propagates from the front to back in each environment. This model moves forward by oscillating its tail much more than the bodies. And the smaller the density of model is, the faster the model oscillates its bodies in the fluid environment, because the creature in the environment whose density is larger needs more energy to move its bodies than that in the environment whose density is smaller. Therefore, the smaller the density of model is, the farer the model moves forward from the start position. In addition, the speed of fish's body generates the drag forces. The speed to unbend model's body ( $\omega_2, \omega_4$ ) is faster than that to bend its body ( $\omega_1, \omega_3$ ) as modeling behavior for flat fish model (Fig.16). Therefore, this model can generate propulsion by applying evolutionary computations (ANN and RCGA).



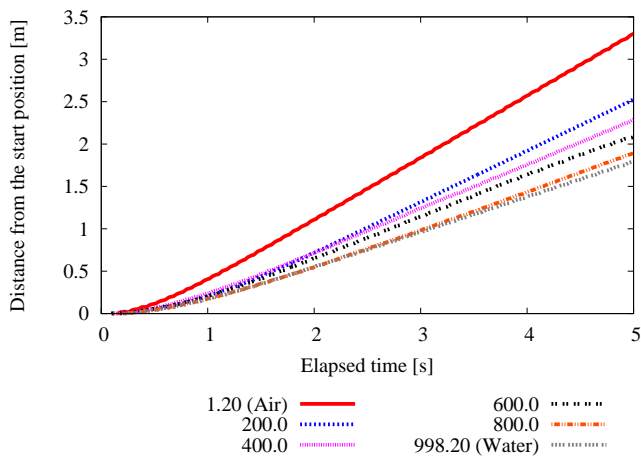


Figure 9: Relation of the fluid environment and the distance from start position

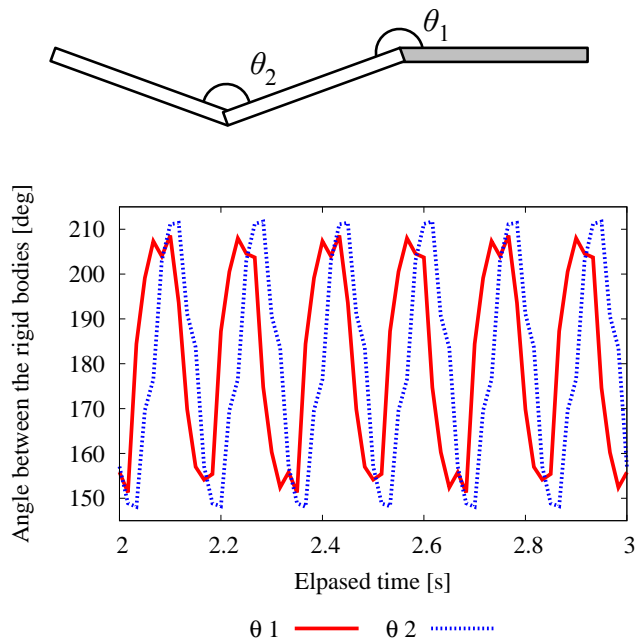


Figure 10: Angles of the rigid bodies on the best individual (Fluid density is the air)

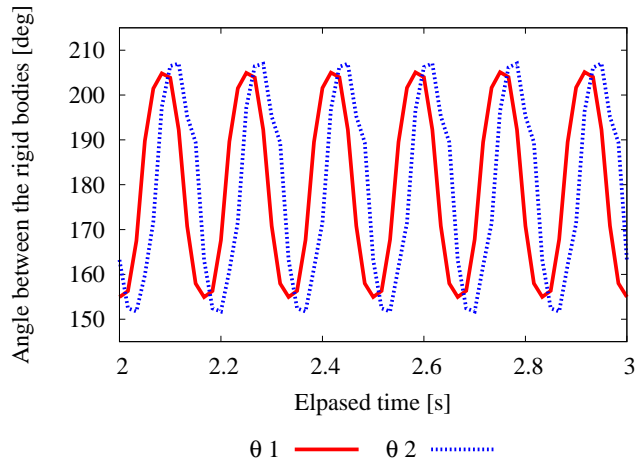


Figure 11: Angles of the rigid bodies on the best individual (Fluid density is 200.0[kg/m<sup>3</sup>])

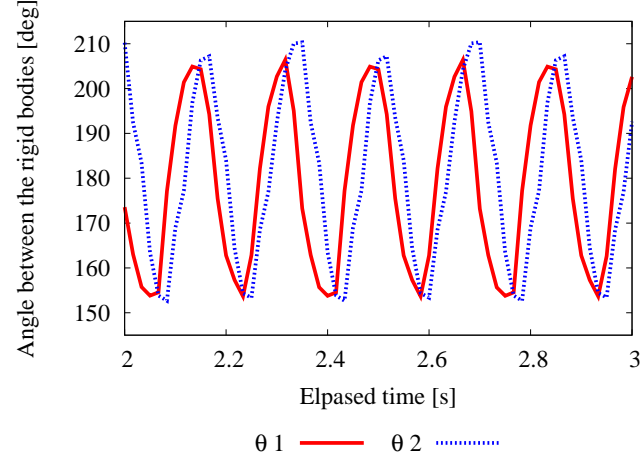


Figure 12: Angles of the rigid bodies on the best individual (Fluid density is 400.0[kg/m<sup>3</sup>])

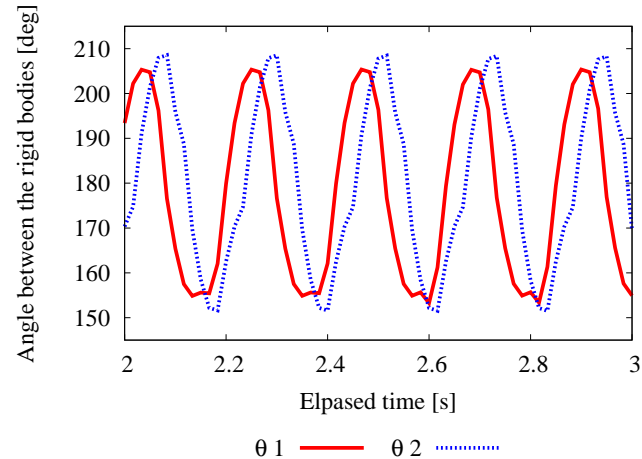


Figure 13: Angles of the rigid bodies on the best individual (Fluid density is 600.0[kg/m<sup>3</sup>])

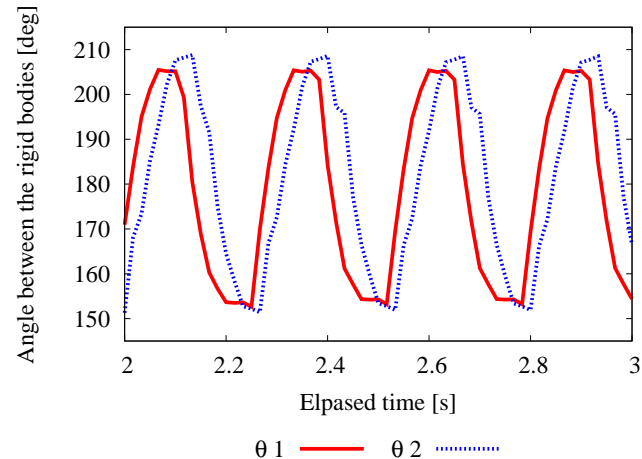


Figure 14: Angles of the rigid bodies on the best individual (Fluid density is 800.0[kg/m<sup>3</sup>])

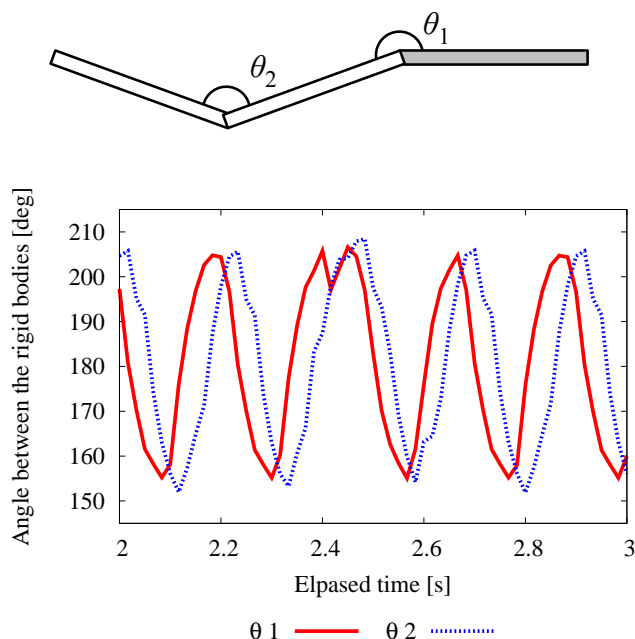


Figure 15: Angles of the rigid bodies on the best individual (Fluid density is the water)

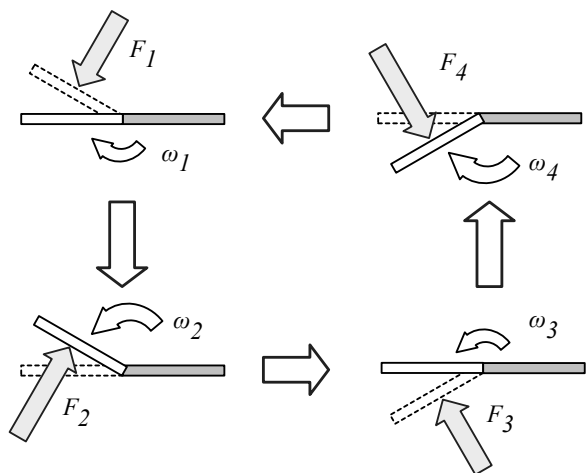


Figure 16: Mechanism generating propulsion

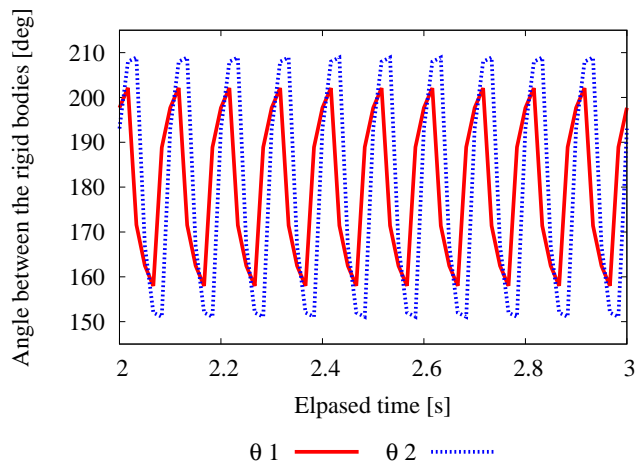


Figure 17: Angle of rigid bodies in the flat model at the 100th generation in the air environment

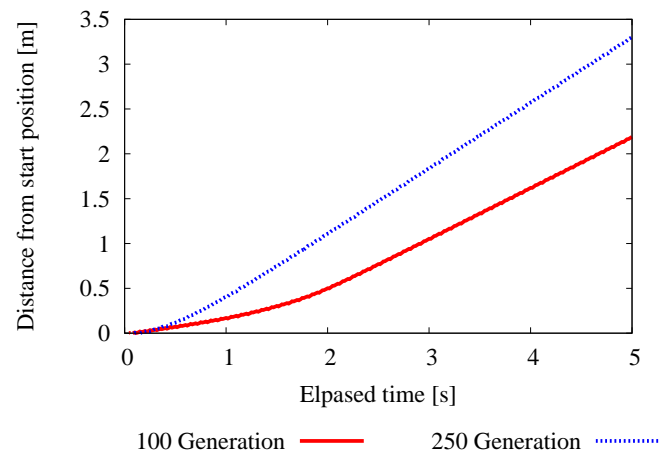


Figure 18: Relation of the generation and the distance from the start position in the air environment

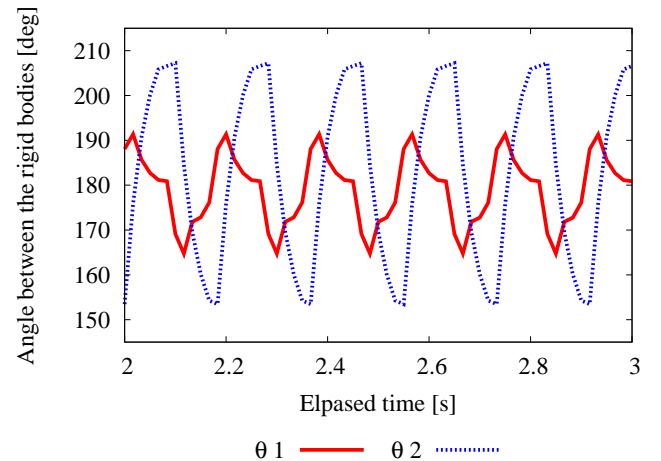


Figure 19: Angle of rigid bodies in the flat model at the 100th generation in the water environment

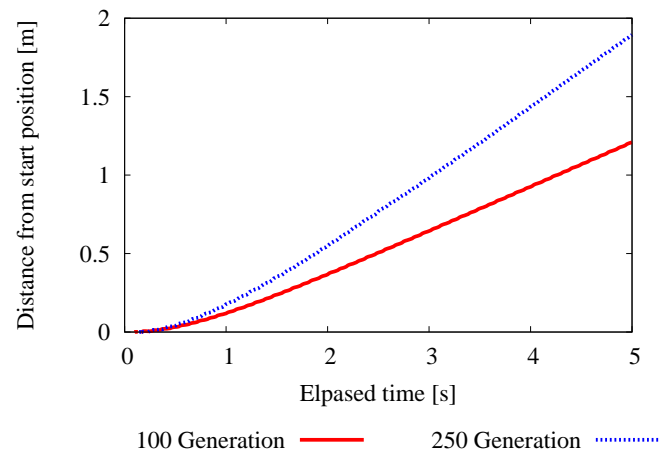


Figure 20: Relation of the generation and the distance from the start position in the water environment

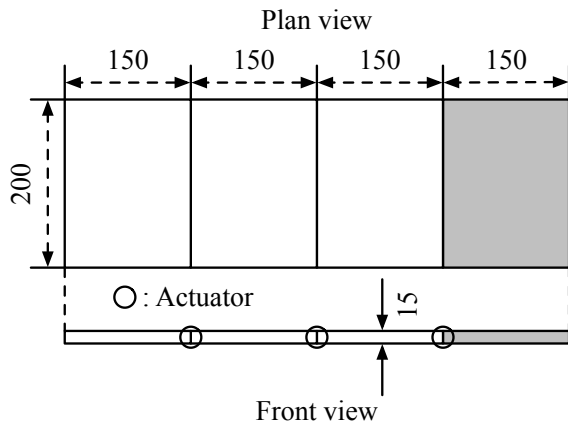


Figure 21: Three Joints Model

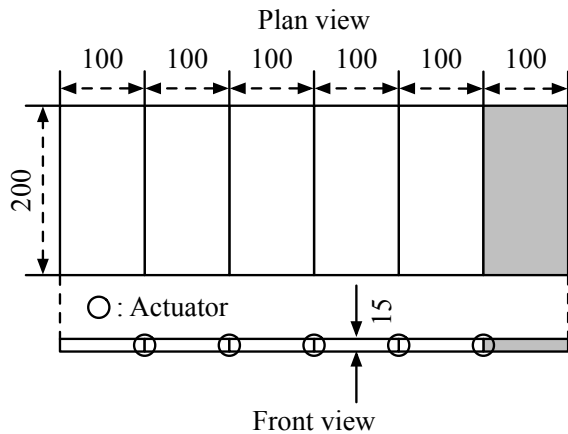


Figure 22: Five Joints Model

In the air environment, the frequency of the elite creature's body in the middle of the experiment is larger than that of creatures's body after optimizing the ANN (Fig.17). However the creature on the way of the experiment cannot move more forward well (Fig.18). Similarly, the creature on the way of the experiment cannot move more forward well in the water environment (Fig.19, 20). This creature acquires an adaptive behavior in the each environment by using evolutionary computations (ANN and RCGA), moves forward as efficiently as possible.

### Additional experiment

Additionally, we examine how the topology of the artificial creature affects with the behavior ability through numerical simulation. To do so, we generate two types of the flat fish model. The modification is done by changing the number of actuators. We make a three joints flat fish model (Fig.21), and a five joints flat fish model (Fig.22). These models consist of rectangular parallelepipeds with the same size, keep-

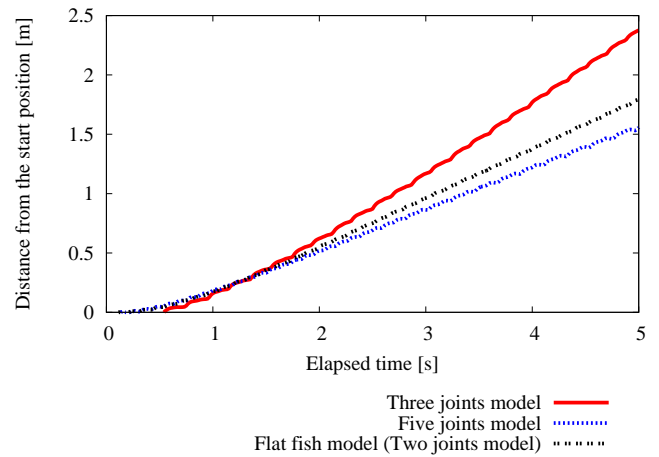


Figure 23: Relation of the number of actuators and the distance from start position

ing the total length of flat fish model (two joint model). We investigate how far two models move forward as efficiently as possible from the initial position within a definite period of time. Evolutionary computation (RCGA) is applied to all generated creatures to adapt their ANNs, which are set as controllers for the behavior. The experimental conditions for the RCGA and the ANN are shown in Table 2. The density of the fluid and model is as same as that of the water.

Figure 23 shows the position of best individual of each model at each simulation time in the water environment. Figure 24 shows the angle between rigid bodies on the best individual of the three joints model. Figure 25 shows the angle between rigid bodies on best individual of the five joints model.

From these results, three joints model move forward further than two joints model from the start position. However, five joints model do not move forward further than two joints model from the start position. Bodies of two models oscillate periodically and the angle between the creature's bodies propagates from the front to back. This creature moves forward by oscillating its tail much more than the bodies like a two joints model. And the speed to unbend each model's body ( $\omega_2, \omega_4$ ) is faster than that to bend its body ( $\omega_1, \omega_3$ ) as modeling behavior for flat fish model (Fig.16).

In addition, the three joints model oscillates its bodies greatly and slowly, This model generates stronger drag forces because the surface drag area is large. On the other hand, the five joints model oscillates its bodies in a small range with a fast frequency. This model generates a small drag forces because the surface drag area is small. By these experiments, it becomes clear that the flat fish model needs a proper topology of the body to move forward, that is, the topology of the flat fish model effects behaviors.

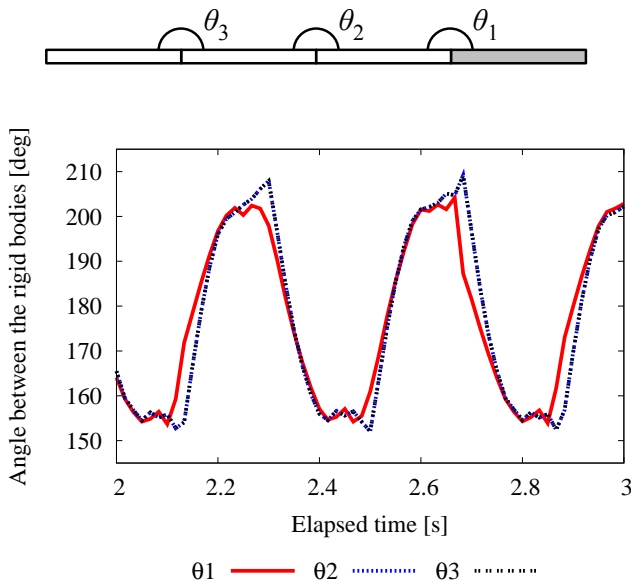


Figure 24: Angles of the rigid bodies on the best individual (three joints model)

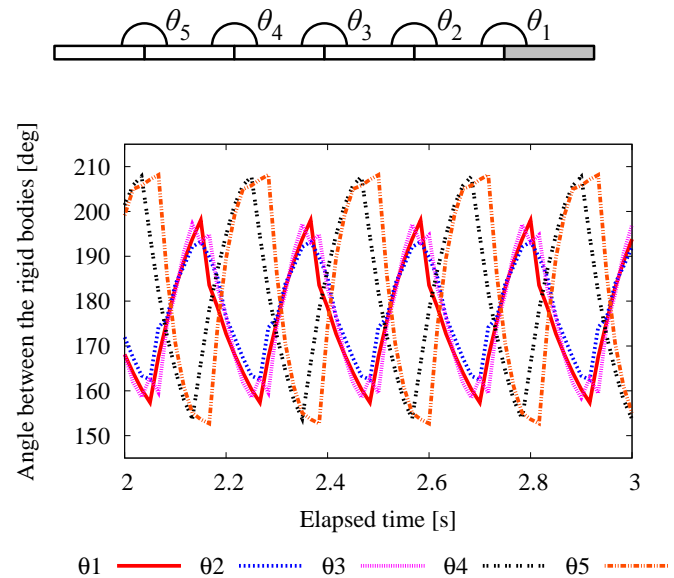


Figure 25: Angles of the rigid bodies on the best individual (five joints model)

## Conclusion

In this paper, we constructed the virtual environment with low computational costs by introducing two forces comparing to the buoyancy and drag calculated by using a physical calculating engine. And we examine how the differences appear when artificial creature model autonomously behaves in some fluid environments by applying evolutionary computing (ANN and RCGA). From the result, it is possible for the model to acquire behaviors in some fluid environment. After optimizing the ANN, this model behaves effectively by leveraging fluid forces in each environment. The model's bodies oscillate periodically, and the angle between the its bodies propagates from the front to back in each environment. This model moves forward by oscillating its tail much more than the bodies. Additionally, we examine how the topology of the artificial creature affects with the behavior ability through numerical simulation. From the result, it becomes clear that the flat fish model needs a proper topology of the body to move forward.

As a future work, we would like to explore "life-as-it-could-be" by controlling the artificial creature which has many wings.

## References

- Chaumont, N., Egli, R., and Adami, C. (2007). Evolving virtual creatures and catapults. *Artificial Life*, 13(2):139–157.
- Koshizuka, S., Nobe, A., and Oka, Y. (1998). Numerical analysis of breaking waves using the moving particle semi-implicit method. *International Journal for Numerical Methods in Fluids*, 26(7):751–769.
- Lentine, M., Grétarsson, J., Schroeder, C., Robinson-Mosher, A., and Fedkiw, R. (2010). Creature Control in a Fluid Environment. *IEEE Transactions on Visualization and Computer Graphics*.
- Miconi, T. (2008). In silicon no one can hear you scream: evolving fighting creatures. In *Proceedings of the 11th European conference on Genetic programming*, pages 25–36. Springer-Verlag.
- Reynolds, C. (1987). Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34. ACM.
- Sims, K. (1994a). Evolving 3D morphology and behavior by competition. *Artificial Life*, 1(4):353–372.
- Sims, K. (1994b). Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 15–22. ACM.
- Tanev, I., Ray, T., and Buller, A. (2005). Automated evolutionary design, robustness, and adaptation of sidewinding locomotion of a simulated snake-like robot. *Robotics, IEEE Transactions on*, 21(4):632–645.
- Tu, X. and Terzopoulos, D. (1994). Artificial fishes: Physics, locomotion, perception, behavior. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 43–50. ACM.
- Usami, Y. (2007). Re-examination of swimming motion of virtually evolved creature based on fluid dynamics. *Advances in Artificial Life*, pages 183–192.

# Autonomous Learning in an Information Stream through Autopoiesis

Nikolaos Nanas<sup>1</sup>

<sup>1</sup>Center for Research and Technology Thessaly, Greece  
n.nanas@cereteth.gr

## Abstract

Nootropia is a complex, self-organizing system, inspired by the Theory of Autopoiesis and successfully applied so far to the challenging problem of profiling a user's information interests. In this paper for the first time, Nootropia is studied in the context of Artificial Life, as an autonomous system that can learn without human intervention. A series of experiments demonstrate that Nootropia can autonomously learn to identify documents belonging to a specific topic with minimal training. This is achieved through a deterministic process of self-organization, which, when coupled with a complex and dynamic information environment, gives rise to rich and unpredictable behavior. Nootropia is open to its environment and operates far from equilibrium, while it tries to maintain its identity within an information stream. Our exploration of the dynamics behind Nootropia's autonomous learning capabilities lead to interesting insights, which may extend beyond its successful application to the problem of profiling and towards a new research stream that uses Nootropia as a means for studying computational autopoiesis.

## Introduction

Humberto Maturana and Francisco J. Varela's *Autopoietic Theory* describes a model of self-organisation (Varela et al., 1974; Maturana and Varela, 1980). In simple words, it states that a system's organisation is defined by its "structure" (its components (nodes) and their relations (links)) and the processes that this structure performs, which continuously regenerate the structure that produces them. Of particular interest to the current work is Varela's view of the immune system in the context of Autopoietic Theory. Varela treated the immune system as an *organisationally closed* network that reacts autonomously in order to define and preserve the organism's identity, in what is called *self-assertion* (Varela and Coutinho, 1991). Self-assertion is an on going process, since both the organism and the environment change over time.

Two types of change contribute to self-assertion. The network's *dynamics* refer to ongoing variations in the concentration of antibodies and play the role of reinforcement learning. The network's *metadynamics* are the result of the recruitment of new cells (produced by the bone marrow)

and of the removal of existing cells. The network's metadynamics play the role of a distributed control mechanism that allows the network to maintain its viability by shifting its immune repertoire (Bersini and Varela, 1994). It is also important, that due to the interactions between antibodies, it is essentially the network itself that chooses which new recruited cells will survive in the network. According to Vaz and Varela, self-assertion is the natural consequence of this *endogenous selection* process (Vaz and Varela, 1978).

Stewart and Varela used a computational model to explore self-assertion (Stewart and Varela, 1991). Like the original computer simulation of a cell-like autopoietic structure in (Varela et al., 1974), Stewart and Varela's model involved a discrete two dimensional grid representation of shape-space, where antibodies are randomly introduced. The survival of antibodies on this grid depends on their affinity to other antibodies, with affinity being a function of the distance between two antibodies. The simulation gave rise to stable (but not static) patterns that were the result of the network's metadynamics. Similar self-assertion models have also been studied in (De Boer and Perelson, 1991) and (Bersini, 2002).

Discrete, two dimensional spaces have been the basis of many computational models of autopoiesis. A comprehensive review can be found in (McMullin, 2004). Although cellular automata on two-dimensional grids are known to be capable of universal computation, in the case of autopoiesis and self-assertion models in particular, the simulated environments are relatively simple. For instance, in the original computational model of autopoiesis, the environment where the cell-like structure is formed comprises particles that bond in the presence of a catalyst to form the cell's membrane. Similarly, in Stewart and Varela's model of the immune system the external environment consists of randomly generated antibodies in the shape space. In both cases, the computer simulations demonstrate visually, that despite the stochastic nature of the environment stable structures progressively emerge and manage to maintain their identity over time.

This paper suggests an alternative scientific methodology

for exploring autonomous behaviour through autopoiesis. It uses the Web as a source of real-world data for simulating a complex and dynamic information environment. In this environment, a profiling system, which has been inspired by the autopoietic view of the immune system, has to autonomously learn to identify specific information, in order to maintain its identity. A series of experiments demonstrates that this system is capable of autonomous learning through a process of self-organisation that dynamically controls the profile's structure. Although, the adopted information environment cannot be easily visualised, depicting how certain macroscopic variables vary over time, reveals a complex system that, although deterministic, is unpredictable. Small variations in the initial conditions can cause significant variations in the structural pathways the system follows as it interacts with its complex environment. The results also reveal an interesting relation between energy consumption and autonomous behaviour that requires further investigation.

### Profiling with Nootropia

According to (Mireille, 2008), profiling could be generally defined as:

“The process of ‘discovering’ correlations between data in databases that can be used to identify and represent a human or nonhuman subject (individual or group) and/or the application of profiles (sets of correlated data) to individuate and represent a subject or to identify a subject as a member of a group or category.”

In practice, when profiling an individual's (or a group's) information interests, a profile is built and continuously learns from the user's interaction with information and is used to evaluate the relevance of new, incoming information to these interests. Profiling in this case, is a challenging problem with analogies to the immune system's self-assertion process. To maintain its viability a profile has to be able to define and preserve the identity of the user's interests. It has to be able to learn a variety of interests and continuously adapt to changes in them.

These analogies inspired the design and development of Nootropia<sup>1</sup>, a profiling system that so far, has been successfully applied for adaptive filtering of textual information according to a user's (or a group's) interests. In its current form, Nootropia was first introduced in (Nanas et al., 2004) and since then, it has been extensively described and experimentally evaluated (see for instance (Nanas and De Roeck, 2009; Nanas et al., 2009, 2010b,a).

In Nootropia, the profile is a weighted network of features, e.g., a network of words extracted from the content of text documents. The links in this network capture correlations between features that appear regularly in the same

context, e.g., correlations between words that appear close to each other in text. A feature's weight measures its importance within the profile and a link's weight the strength of the correlation between two features. The profile is built and continuously adapts to interest changes through a process of self-organisation that adjusts the network's structure in response to user feedback (explicit or implicit). For instance, if a document is identified as relevant to the user's interests, then words in the profile that also appear in the document get reinforced at the expense of the words they are linked to. These local competitions cause a redistribution of weight between the profile's words (dynamics). Words in the document that do not already appear in the profile are recruited and those profile words that run out of weight are purged (metadynamics). The exact self-organisation process is described in detail in (Nanas and De Roeck, 2009).

To evaluate the relevance of an information item (e.g., document), the profile deploys a directional spreading activation process. Profile features (e.g., words) that also appear in the item get activated. In order of increasing weight, each activated feature disseminates part of its current activation towards the activated features with larger weights that it is linked to. The relevance score is then calculated as the weighted sum of the final activation of profile features. This non-linear evaluation process, which is described in detail in (Nanas et al., 2010a), implies a hierarchy of features, as activation is being channeled from the majority of features with small weights towards the “elite” of features with large weights. The structure of this implicit hierarchy, which continuously self-organises in response to the environment, defines the profile's collective reaction to incoming information.

The autopoietic properties of Nootropia are discussed in detail in (Nanas and De Roeck, 2009), where it is argued that Nootropia exhibits the basic characteristics of self-assertion models. It is a non-linear, self-organising system, that is open to its environment and operates far from equilibrium, constantly adjusting structurally, and hence behaviourally. It also involves both network dynamics and metadynamics with endogenous selection. Experiments performed in (Nanas and De Roeck, 2009) and (Nanas et al., 2010b) demonstrate Nootropia's ability to effectively adapt to a variety of interest changes through self-organisation. Further experiments and analysis indicate that it is the network's non-linearity which allows the profile to store additional information regarding a user's interests and thus remain specific even within high-dimensional spaces (Nanas et al., 2010b,a). In such spaces, comparative experiments between Nootropia and a vector-based profile containing the same weighted words, show that the additional information encoded by Nootropia's links contributes to an increase in accuracy of up to 50% (Nanas et al., 2010a). Nootropia's advantageous properties have already boosted the development of real world prototypes, such as the Personalised

<sup>1</sup>Greek word for: “an individual's or a group's particular way of thinking, someone's characteristics of intellect and perception”.



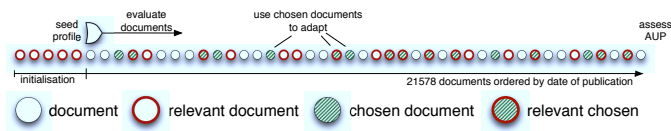


Figure 1: Experimental Process

News Aggregator described in (Nanas et al., 2010c).

In all of the above cases though, it is the user (or a group of users) that explicitly, or implicitly, provides the profile with relevant information to learn from. So what if we take the user out of the equation and ask the profile to autonomously identify and choose information to learn from? Will the profile be able to maintain its identity and what lessons can be learned from its autonomous learning behaviour? This paper deals with these questions experimentally in the context of Alife in general and of the Theory of Autopoiesis in particular.

## Experimental Evaluation

The performed experiments are a continuation of those referred above and use a variation of their methodology to test the ability of a profile to autonomously learn to identify documents belonging to a specific topic category. Once more the dataset used in the experiments is the Reuters-21578 document collection<sup>2</sup>. It includes 21578 news stories from Reuters newswire in 1987, ordered according to publication date and classified by human experts into 135 topic categories. The experiments focus on the 23 topics with at least 100 relevant documents in the dataset.

### Autonomous Profile

As it is exemplified in figure 1, for each of the 23 topics, a “seed” profile is initialised using the first five<sup>3</sup> documents in the collection belonging to the topic. The seed profile is then released in the information stream and traverses the 21578 documents in the collection in chronological order. The profile evaluates every individual document and assigns to it a score. If the assigned score is over a threshold the profile chooses the document for “consumption” and self-organises accordingly. For the current experiments, the threshold is calculated for each individual document as the average score assigned to the documents “consumed” so far. The process is repeated until all 21578 have been accounted for. The

<sup>2</sup>available at <http://www.daviddlewis.com/resources/testcollections/reuters21578/>

<sup>3</sup>Experiments were also performed for 1, 10 and 50 initialisation documents, but are not reported here due to space limitations. With just 1 initialisation document the seed profile is not developed enough to achieve the desired behaviour. As the number of initialisation documents increases from 5 to 10 and then to 50 the profile relies more on its initial condition rather than the subsequent learning process.

profile’s accuracy is then measured by calculating the Average Uninterpolated Precision (AUP) of the list comprising the documents in the collection ordered by decreasing score. A topics AUP is defined as the sum of the precision<sup>4</sup> at each point in the ordered list where a relevant document appears, divided by the total number of relevant documents. The essence of this accuracy metric is that documents relevant to the current topic should receive larger scores than irrelevant documents.

The above methodology establishes a challenging experimental task. Based only on its initial training with a small number of documents relevant to a topic, the profile has to autonomously learn to identify documents belonging to that topic. Ideally, the profile should choose all relevant documents and ignore the rest<sup>5</sup>. However, as it is depicted in figure 1, there are typically both false negatives and false positives. Not all relevant documents are chosen and not all chosen documents are relevant. Both the percentage of relevant documents chosen and the percentage of chosen documents that are relevant affect the profile’s accuracy. If the first percentage is small the profile ignores valuable input. If the second percentage is small then the profile may deviate away from the current topic of interest. It should also be noted that since the content of documents relevant to a topic may change over time, the profile has to be able to follow this drift. Overall, the choices the profile made so far define its current structure and consequently its future choices. So even small changes in the initial conditions can cause the profile to follow a very different trajectory. Out of an infinite number of possible network configuration the profile has to self-organise in such a way that it manages to maintain its (topical) identity within a complex and dynamic environment.

### Supervised and Random Profiles

In the experiments the accuracy and behaviour of the autonomous profile are juxtaposed with those of a *supervised profile* and of a *random profile*<sup>6</sup>. In both cases we start with an initially empty profile. Like before the profile is released in the information stream and evaluates the 21578 documents in chronological order. Unlike the autonomous profile, these two types of profile do not choose the documents to learn from autonomously. Whenever the supervised profile evaluates a relevant document it will always use it for learning, while it ignores all non-relevant documents. In other words, it is provided a priori with complete knowledge of which documents are relevant to the current topic of

<sup>4</sup>i.e., the ratio of documents relevant to that topic.

<sup>5</sup>It is assumed that the categorisation of documents by Reuter’s experts has been accurate.

<sup>6</sup>All three types of profile are built using Information Gain to extract the most important words in the training documents and a sliding window of size 20 for identifying correlations between the extracted words.



topic code	relevant docs	docs chosen	relevant chosen	rel. chosen/ docs chosen	rel. chosen/ total rel.	AUP autonomous	AUP supervised	AUP random	auto/ supervised
earn	3987	715	694	0.97	0.17	0.694	0.732	0.349	0.949
acq	2448	433	149	0.34	0.06	0.262	0.424	0.105	0.617
money-fx	801	16	11	0.69	0.01	0.311	0.556	0.046	0.559
crude	634	123	114	0.93	0.18	0.636	0.700	0.033	0.909
grain	628	6	1	0.17	0.00	0.246	0.509	0.027	0.483
trade	552	94	65	0.69	0.12	0.334	0.558	0.044	0.599
interest	513	124	108	0.87	0.21	0.413	0.463	0.030	0.892
wheat	306	77	53	0.69	0.17	0.430	0.490	0.012	0.878
ship	305	54	6	0.11	0.02	0.029	0.436	0.011	0.066
corn	254	8	3	0.38	0.01	0.322	0.275	0.010	1.171
dlr	217	97	52	0.54	0.24	0.371	0.468	0.014	0.793
oilseed	192	7	2	0.29	0.01	0.298	0.174	0.010	1.714
money-supply	190	719	64	0.09	0.34	0.051	0.184	0.012	0.279
sugar	184	199	30	0.15	0.16	0.116	0.683	0.008	0.169
gnp	163	8	3	0.38	0.02	0.384	0.424	0.013	0.907
coffee	145	51	45	0.88	0.31	0.775	0.824	0.007	0.940
veg-oil	137	182	21	0.12	0.15	0.082	0.459	0.012	0.180
gold	135	10	5	0.50	0.04	0.763	0.768	0.005	0.993
nat-gas	130	15	9	0.60	0.07	0.665	0.432	0.007	1.538
soybean	120	7	2	0.29	0.02	0.421	0.285	0.005	1.479
bop	116	230	45	0.20	0.39	0.175	0.310	0.005	0.566
livestock	114	8	3	0.38	0.03	0.111	0.265	0.006	0.419
cpi	112	119	26	0.22	0.23	0.065	0.285	0.005	0.229
average	538.4	143.6	65.7	0.5	0.1	0.346	0.465	0.034	0.743

Table 1: Experimental Results. Columns from left to right: (1) topic code, (2) number of relevant documents in the collection, (3) number of documents chosen by the the autonomous profile, (4) number of chosen documents relevant to the current topic, (5) ratio of chosen documents that are relevant, (6) ratio of relevant documents chosen, (7) per topic AUP score for the autonomous profile, (8) per topic AUP score for the supervised profile, (9) per topic AUP for the random profile, (10) ratio of the supervised profile’s AUP achieved by the autonomous profile.

interest. The random profile, on the other hand, is provided with an equal number of randomly selected documents from the collection.

## Accuracy

Table 1 summarises for each topic, the choices made by the autonomous profile and the resulting AUP score and compares it to those of the supervised and autonomous profile. The results lead to the following observations:

- The accuracy of the random profile is the lowest (table 1 col. 9). The profile must learn from relevant documents to be accurate.
- As expected the supervised profile achieves the best overall performance (table 1 col. 8).
- The performance of the autonomous profile is satisfactory (table 1 col. 7). It achieves on average 74% of the supervised profile’s accuracy (table 1 col. 10).
- The autonomous profile achieves this level of accuracy although on average it only identifies 10% of the existing relevant documents per topic (table 1 col. 6). It appears that not all of the available relevant documents are required for increased accuracy. In fact, it is interesting that there are four topics (corn, oilseed, nat-gas, soybean) for which the autonomous profile clearly outperforms the supervised profile although, after its initialisation with five documents, it chooses a very small number of documents to learn from. It may be the case, that for certain topics with relatively small number of relevant documents in

the collection and distinct content, this is a better strategy. The seed profile overspecialises to the initialisation documents, but these are representative enough of the remaining relevant documents that ignoring them leads to better accuracy. In any case, this is not always the best strategy (e.g., topics grain, ship, and livestock).

- The satisfactory accuracy of the autonomous profile is mainly due to the fact that, on average, 50% of the documents chosen are indeed relevant. There is a clearer correlation between the profile’s accuracy and the percentage of chosen documents that are relevant. In general, if the percentage is small the accuracy of the autonomous profile is small and increases as the percentage increases. For percentages close to one the accuracy of the autonomous profile approximates that of the supervised profile. It is clear that the profile has to be selective when choosing the documents to learn from. Too many false positives can cause the profile to drift away from the current topic of interest.

## Behaviour

Nootropia is a complex system and it is not easy to visualise, or to analyse, its dynamic behaviour. In this paper, an attempt is made to understand how self-organisation contributes to the above autonomous learning capabilities, by observing certain macroscopic variables related to the profile’s nodes and their weights. The analysis of the network’s connectivity is part of ongoing work and will be included in future publications. Furthermore, due to space limitations,

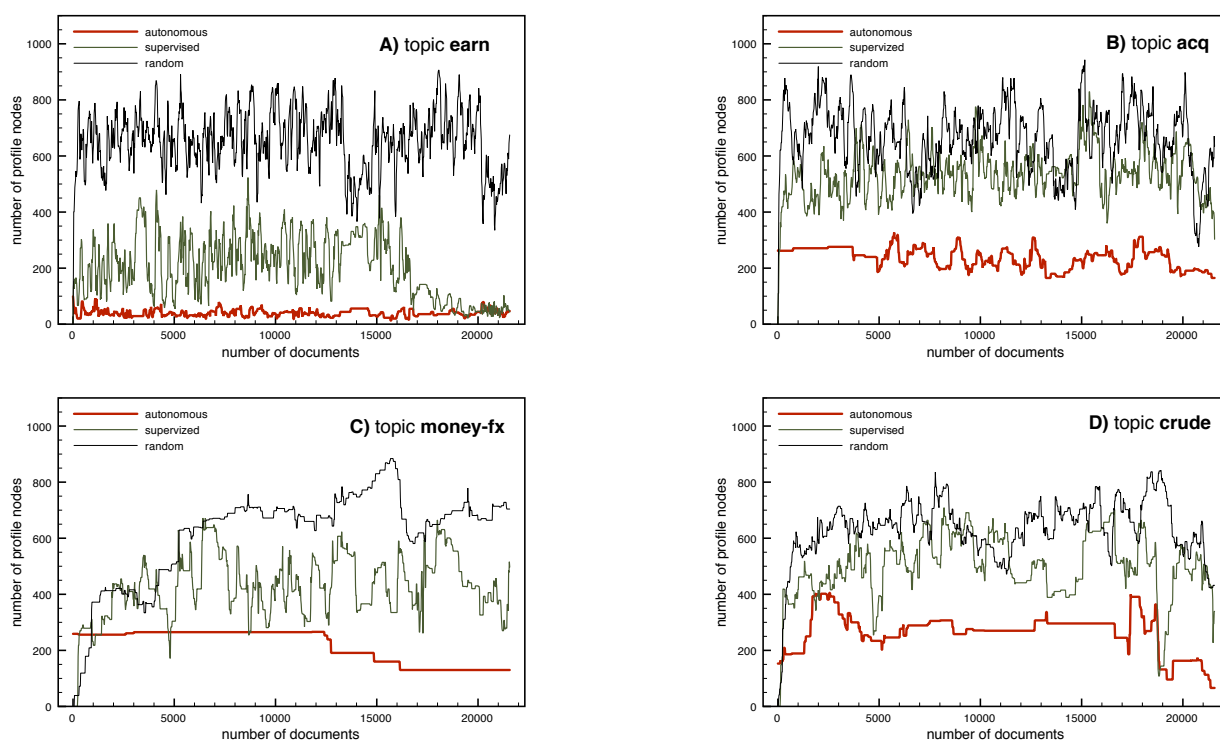


Figure 2: Fluctuations in the number of profile nodes along the information stream.

only the four topics with the largest number of relevant documents are indicatively chosen for this study.

The graphs in figure 2 depict for each of the topics earn, acq, money-fx and crude, how the number of nodes (words) in the profile (Y-axis) changes as it traverses the document collection (X-axis). These indicative graphs show that Nootropia is a dissipative, self-organising system that can dynamically control its size (and connectivity (Nanas et al., 2009)). Energy (word weight) flows through the profile with the addition of words and is dissipated when these words are purged. Although there are more than 20.000 unique words in Reuters-21578<sup>7</sup>, the number of nodes in the profile does not escalate above 1000. In all four cases, the three types of profile are easily distinguished based on the average number of words. The autonomous profile maintains the smallest number of words and the random profile the largest number of words, although it uses the same number of documents to learn from as the supervised profile. So these differences are not only due to differences in the number of training documents, but they also depend on the semantic diversity of these documents. The random profile is provided with randomly selected training documents from the collection, that may belong to any topic. These documents may include a greater variety of words and thus give rise to a profile

with a larger number of nodes. The supervised profile uses the same number of documents relevant to a specific topic and so their vocabulary is more focused. For the same reasons, the autonomous profile appears to be the most focused profile type, with the least number of profile words and the mildest fluctuations. Apparently, the profile has the ability to choose documents that are semantically close to its initial composition and their vocabulary is already reflected in the profile. These documents do not have many new words to contribute to the profile and cause as a result smaller profile perturbations. It is also evident from these figures that the average number of words in each profile type varies from topic to topic and depends not only on the number of relevant documents, but also on the semantic characteristics of each topic. Finally, it is clear that in the case of topic money-fx (fig. 2 C ), the profile does not successfully identify appropriate documents to learn from, causing a decrease in the number of profile words and the poorest relative accuracy out of the four cases (see tbl. 1).

To further investigate the behaviour of Nootropia, figures 3 and 4 depict respectively, the average and aggregate weight of profile nodes through out the 21578 documents<sup>8</sup>. With the exception of the unsuccessful topic money-fx, the autonomous profile has the largest average weight, which

<sup>7</sup>After stop word removal and stemming.

<sup>8</sup>Note that for visualisation reasons the Y-axis of the graphs in figure 3 has various scales.

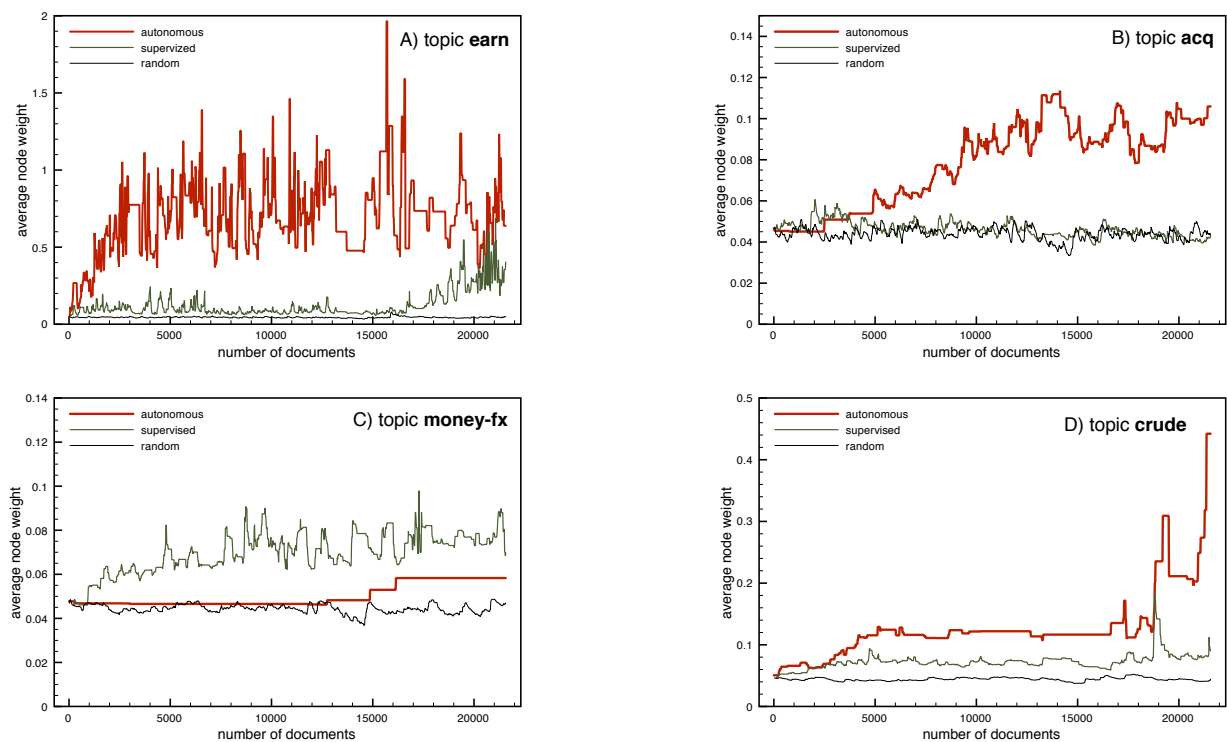


Figure 3: The average weight of profile nodes along the information stream.

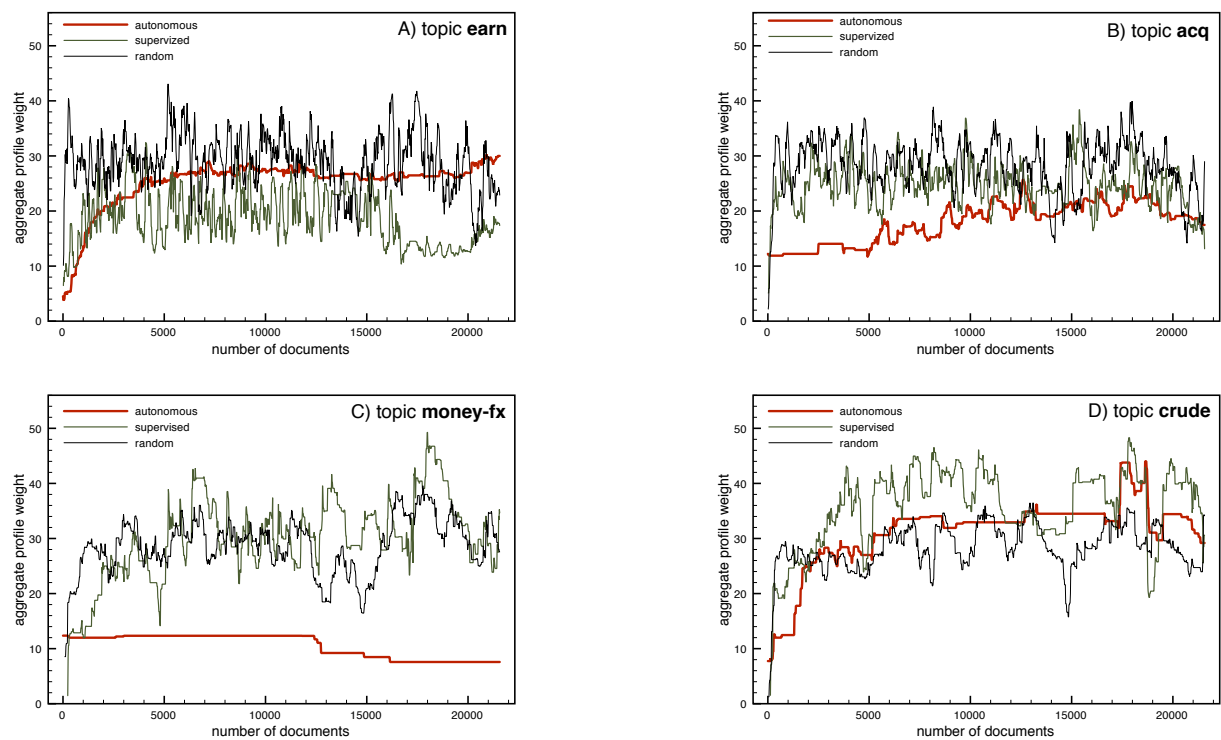


Figure 4: The aggregate profile weight along the information stream.

tends to increase along the process. There is also an apparent correlation between the scale of the average node weight and the accuracy of the profile (see table 1). Furthermore, the average weight does not depend only on the number of profile nodes. According to graphs B and D in figures 2 and 3, although the autonomous profile maintains in both cases approximately the same number of nodes, there is a significant difference in the average weight. This means that the average weight of profile nodes is possibly another macroscopic variable that characterises the behaviour and accuracy of the autonomous profile. It shows that the autonomous profile can effectively maintain and reinforce its identity. By choosing documents that are relevant to its initial semantic composition, the profile reinforces what has already been learned and remains specific to its area of interest, thus avoiding intense structural fluctuations.

The distinct behaviour of the autonomous profile is also reflected in the way it aggregates node weight. According to figure 4, with the exception again of topic money-fx, the autonomous profile progressively accumulates weight until it reaches a certain capacity, where it tends to stabilise. It is also interesting, that unlike the supervised and the random profile, the aggregation of weight by the autonomous profile is more progressive and with less fluctuations. Finally, it is notable that for topic earn, acq and crude the aggregate weight of the autonomous profile is comparable to that of the supervised and random profiles, despite the smaller number of documents used for learning (tbl. 1) and the smaller number of profile words (fig. 2).

## Discussion

The experimental results show that Nootropia is capable of autonomous learning within a complex information environment. The system's accuracy in itself is not the primary concern of this paper. It is already satisfactory enough, given the small amount of training data that are provided for initialisation and it can be further improved, e.g., through more elaborate thresholding mechanisms. What is important is that this unsupervised learning behaviour is the result of an autopoietic network's self-organisation in response to a diverse and changing information environment. The network "perceives" its environment through a non-linear spreading activation process that leads to increased specificity even within high-dimensional environments (Nanas et al., 2010a). As a result, the network can accurately identify and extract relevant information from the environment. The "cognitive", learning process is the result of the network's reaction to the extracted information and involves both the redistribution of node weights through local interactions (network dynamics) and the addition and removal of nodes (network metadynamics). The network becomes open to the environment: energy (weight) is absorbed from the environment, it is temporarily stored by the network and eventually, it is disseminated back to the environment. The distribution of

stored energy (weight) in the network imposes a hierarchy on the nodes that defines the network's response to the environment. When the network is forced to self-organise in response to random information then it becomes large (fig. 2), but the hierarchy of nodes remains shallow (fig. 3). The network is more volatile, because more nodes have small weights and can be more easily removed from the network, causing pronounced fluctuations in the number of nodes. On the contrary, relevant information reinforces what is already in the network with additional energy and the hierarchy of nodes grows higher. This increases the stability and specificity of the network and it becomes more likely that it will identify additional relevant information, leading to a positive feedback loop, which allows the profile to maintain its identity and to avoid strong perturbations.

Some interesting lessons can be learned from all the above:

- The World Wide Web can serve as a valuable source of real-world data, for simulating complex and dynamic environments to experiment in the domain of Artificial Life. These environments are multidimensional and cannot be visualised. They provide however a rich information world that lies somewhere in the middle of the range between the relatively simple 2D worlds of many computer simulations and the physical world. As in the case of Varela's 2D simulations (Varela et al., 1974; Stewart and Varela, 1991), the above experiments demonstrate that even in such a complex environment autopoiesis can still give rise to consistent, "meaningful" behaviour that can maintain a system's viability.
- If Nootropia is indeed an autopoietic system, or at least exhibits some autopoietic properties, then its experimental study highlights the importance of the environment during autopoiesis. Nootropia is organisationally closed, but it is the interaction with the environment that guides its structural and hence, behavioural development. It is structurally coupled to its environment and unlike existing 2D simulations, it is the richness of this environment that can give rise to a plethora of structural modifications and corresponding behaviours.
- It is even more interesting that the study of Nootropia's behaviour indicates a relation between energy and autopoiesis. The autonomous profile effectively accumulates energy per node, to reinforce its structure and hence its identity. As a consequence, even within a complex environment and despite the infinite number of possible structural pathways, the profile can be specific enough to choose the information that will lead to further energy aggregation and self-assertion. The role of energy during autopoiesis has been ignored by existing computational investigations and will become a major theme of the research endeavour that this paper initiates.

## Summary and Outlook

Computational Autopoiesis is already an established area of research in Alife. The most common approach involves simulating autopoietic (cell-like) structures on a discrete, two-dimensional space. The current work deviates from this practice. Nootropia is a profiling model that has been inspired by Varela's view of the immune system as an organisationally closed network of interacting antibodies, which reacts autonomously to define and preserve the host organism's identity. Nootropia has already been evaluated extensively and has produced significant results, both quantitatively and qualitatively. The past experimental work on Nootropia concentrated on supervised learning. In this paper for the first time, human intervention is kept minimal. A collection of news articles ordered according to publication date serves as an information stream and within this stream a profile, which has been initialised with a small number of articles relevant to a topic, has to autonomously identify and learn from more relevant articles. This autonomous profile is contrasted to a supervised profile that has complete knowledge of what is relevant and a random profile that chooses documents to learn from at random.

The accuracy of the autonomous profile is satisfactory. It clearly outperforms the random profile and achieves a level of accuracy that is on average 74% that of the supervised profile. What is important is that this level of accuracy is the result of self-organisation in response to the environment. The analysis of Nootropia's behaviour provides evidence that it can control its structure dynamically and in such a way that it effectively consumes and stores energy from the environment. The stored energy reinforces the network's structure and hence the profile's specificity. It becomes easier for the profile to identify more relevant information, leading naturally to self-assertion.

The experimental work in this paper demonstrates also that exploiting the web as a valuable source of real-world data for simulating complex and dynamic environments, can be a fruitful avenue of research in Alife. It is in such multidimensional information worlds that interesting complex structures and behaviours may arise as a natural consequence of autopoiesis. This paper is only a first step in this research avenue. Future steps involve a more extensive experimentation and analysis, including statistical analysis of the network's properties (e.g., degree distribution and clustering coefficient), but also, a more comprehensive exploration of the background theories and philosophies. This paper brings to our attention the relation between information, energy and life and creates one more connection between Alife and the domains of Thermodynamics and Energetics in general. Nootropia is a possible means for exploring this relation through computational autopoiesis and some interesting insights have been gained with the current work. Much more is of course required to be able to make bolder claims, or to draw more general conclusions.

## References

- Bersini, H. (2002). Self-assertion versus self-recognition: A tribute to Francisco Varela. In *ICARIS 2002: 1st International Conference on Artificial Immune Systems*, pages 107–112. University of Kent at Canterbury Printing Unit.
- Bersini, H. and Varela, F. (1994). The immune learning mechanisms: Reinforcement, recruitment and their applications. In *Computing with Biological Metaphors*, pages 166–192. Chapman Hall.
- De Boer, R. J. and Perelson, A. S. (1991). Size and connectivity as emergent properties of a developing immune network. *Journal of Theoretical Biology*, 149:381–424.
- Maturana, H. R. and Varela, F. J. (1980). *Autopoiesis and Cognition: the Realization of the Living*, volume 42 of *Boston Studies in the Philosophy of Science*. D. Reidel Publishing Company, Dordrecht, Holland.
- McMullin, B. (2004). 30 years of computational autopoiesis: A review. *Artificial Life*, 10(3):277–295.
- Mireille, H. (2008). *Defining Profiling: A New Type of Knowledge*, chapter 2, pages 303 – 326. Cross-Disciplinary Perspectives. Dordrecht Springer.
- Nanas, N. and De Roeck, A. (2009). Autopoiesis, the immune system and adaptive information filtering. *Natural Computing*, 8(2):387–427.
- Nanas, N., Uren, V., and De Roeck, A. (2004). Nootropia: a user profiling model based on a self-organising term network. In *Artificial Immune Systems, Third International Conference (ICARIS 2004)*, pages 146–160. Springer, Heidelberg, Germany.
- Nanas, N., Vavalis, M., and De Roeck, A. (2010a). A network-based model for high-dimensional information filtering. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 202–209, New York, NY, USA. ACM.
- Nanas, N., Vavalis, M., and De Roeck, A. (2010b). Words, antibodies and their interactions. *Swarm Intelligence*, 4(4):275–300.
- Nanas, N., Vavalis, M., and Houstis, E. (2010c). Personalised news and scientific literature aggregation. *Information Processing and Management*, 46:268–283.
- Nanas, N., Vavalis, M., and Kellis, L. (2009). Immune learning in a dynamic information environment. In *Artificial Immune Systems, 8th International Conference (ICARIS 2009)*, pages 192–205. Springer, Heidelberg, Germany.
- Stewart, J. and Varela, F. J. (1991). Morphogenesis in shape-space, elementary meta-dynamics in a model of the immune network. *Journal of Theoretical Biology*, 153:477–498.
- Varela, F. J. and Coutinho, A. (1991). Second generation immune network. *Immunology Today*, 12(5):159–166.
- Varela, F. J., Maturana, H., and Uribe, R. (1974). Autopoiesis: The organization of living systems, its characterization and a model. *Biosystems*, 5(4):187–196.
- Vaz, N. M. and Varela, F. (1978). Self and non-sense: An organism-centered approach to immunology. *Medical Hypotheses*, 4:231–267.

# Embodied copying for richer evolution

Adam Nellis<sup>1</sup> and Susan Stepney<sup>1</sup>

<sup>1</sup>York Centre for Complex Systems Analysis (YCCSA), University of York, UK, YO10 5GE  
adam@cs.york.ac.uk

## Abstract

We address the process of *copying* in Artificial Life organisms. Copying is a source of mutations, a crucial component in evolution. We propose that rich copying mechanisms, and thereby rich evolutionary systems, can be obtained by *embodying* the copying process in a lower-level simulation.

We demonstrate an embodied copying process that has the potential to alter its own mutation rate, without having the concept of a mutation rate parameter explicit in the system.

## Introduction

In computing, the concept of copying is important. Many programs copy data during computation. So programming languages often have the concept of copying as a primitive instruction. For example, all high-level imperative languages have an assignment operator;  $a := b$  copies the contents of  $b$  and puts the result in  $a$ .

In Artificial Life (ALife), the concept of copying is also important. To reproduce, life-forms (whether biological or artificial) need to copy themselves. ALife organisms in computers can make use of the copy operations in programming languages, using these to copy themselves. But the requirements of ALife organisms and traditional computer programs are different. Copying in ALife is a source of mutations. It is a novelty-generation process driving evolution.

In biological organisms, copying is not an abstract concept implemented by a defined instruction. It is an emergent property of lower level processes. Copying is *embodied* within the biological systems that are being copied, and so mutations caused by the copying process can change the copying process.

We propose that ALife organisms should not blindly use the copy operations provided by programming languages. Here, we focus on copying as an embodied *process*, rather than as a computational *result*.

Artificial Chemistry (AChem) is the medium we use to embody the copying process. We explain how existing work has started to implement embodied copying reactions in AChem. We build on this by designing an AChem and using it to implement an embodied copying process.

## Crisp, stochastic, and embodied copying

In normal computer programs, copying should happen *crisply*, without any errors. If a programmer writes  $a := b$  in their code, they expect the copy to work perfectly. They expect  $a$  to contain an exact copy of  $b$ .

However, this is not the case in ALife. When biological life-forms (such as bacteria) clone themselves asexually, the clones are not exact copies of their parents (see any biology textbook, e.g. [1]). The biological ‘copy operation’ does not work perfectly. But this is not a mistake. Biology would not be improved by a perfect copy operation. Imperfect copying in biology causes the mutations and novelty that allow evolution to happen.

*Stochasticity* is a way of introducing variation into computer programs (or more generally, any systems). ALife organisms can use this variation to explore the design space of possible organisms. Stochastic programs are crisp programs with variation introduced via pseudo-random number generators. Stochastic programs can influence ALife organisms, allowing the organisms to vary. But the variation originates outside the simulation of the organisms, so the organisms can not influence the variation process. They can not change the stochastic programs. If the programs are to be changed during a simulation, they must be changed by another abstract process, operating on a higher level. This process, in turn, can only be changed by a process operating on an even higher level. This chain of meta-processes and meta-parameters can be broken by *embodying* the process in the simulation.

*Embodying* means implementing one system (the process) within another (the environment). It is frequently used in robotics to refer to building physical robots rather than simulated ones, thus embodying the robot system (process) in the physical world (environment). But the environment within which a process can be embodied is not limited to the physical world [6]. All processes are embodied within some environment, but stochastic programs are embodied in a trivial environment outside the simulation of the ALife organisms. This is why the ALife organisms can not change the stochastic programs.

---

**Algorithm 1** Deconstructing the string copy operation, as a prerequisite to embodying it

---

```
result string_A := string.B

i := start(string.B)
while i not at-end(string.B) do
  string_A(i) := char-copy(string.B(i))
  i := next(i)
end while
```

---

In order for an ALife organism to change a stochastic process, the process must be implemented in the same language as the organisms: the process must be embodied within the simulation of the organism. A stochastic copying process allows the organisms in a simulation to vary, and so evolve. If the copying process is itself part of the simulation, then it too will be able to vary and evolve.

### Copying as a process

In writing an embodied copying program, we must think about the *process* of copying a string, rather than the *result* of the copy. Algorithm 1 breaks down this process into four parts, each involving a particular function: *start*, *at-end*, *char-copy*, and *next*. Each of these four functions can be either crisp, stochastic or embodied. If all four are crisp, then the overall copying process is crisp, and exact copies are always produced.

If any of these four functions are stochastic, then the overall copying process will be stochastic. Making different combinations of these four functions stochastic introduces different kinds of variation into the copying process. For example: making *char-copy* stochastic could cause some characters to be copied incorrectly; making *at-end* stochastic could cause the copy to be truncated.

We can embody the copying process in different ways, and to different degrees. We must implement a simulation of a system where at least one of these functions can happen as a consequence of lower-level events. But we do not need to embody all four of the functions. We can implement some of them as crisp or stochastic functions in the definition of our simulation. Thus there are many different ways in which we can embody the copy operation. Each of these ways leads to different systems with different properties and different degrees of self-modification and novelty generation.

### Example: the Stringmol AChem

The Stringmol AChem [3, 2, 4] has been used to implement an embodied copy operation. In terms of algorithm 1, it has embodied *start* and *at-end* functions, a stochastic *char-copy* function and a crisp *next* function.

Stringmol's embodied copying process has been shown to produce interesting behaviour [3]. Because the process of copying is embodied in a 'replicase' chemical, evolution

can change the process when the replicase copies (another instance of) itself. One sequence of changes observed in Stringmol (described in detail, in [3]) is the emergence of an unprogrammed 'macro-mutation' that chops off the first few characters of a chemical. The emergence of the macro-mutation exploited the two embodied stages of Stringmol's copying process: the *start* and *at-end* functions.

Stringmol produced something different from what would normally be expected of a copy operation: an unprogrammed type of mutation. The emergence of a new type of mutation is not possible using just a stochastic copy operation. Embodiment is needed to allow the intermediate stages of the copying process to be exploited and changed. This shows the potential power of embodying the copying process (or more generally, any process).

Our hypothesis is that by embodying different stages of the copying process, we will be able to observe different, unprogrammed types of mutation emerging from our ALife simulations.

## The Graphmol AChem

In our Graphmol AChem, the chemicals are graphs, and reactions change the topology of the graphs. We use Graphmol to build an embodied copy operation that has an embodied *next* function. Here we use crisp *start*, *at-end* and *char-copy* functions, because we are interested in investigating the effect of embodying the *next* function. However, Graphmol has been designed so that the *start*, *at-end* and *char-copy* functions can (in the future) be made stochastic or embodied.

We embody the *next* function by building a "walker" chemical in Graphmol. This chemical is a graph that can change its own topology by running short computer programs. Some of its graph nodes are "feet" that walk along the string being copied (which is also represented as a graph). The *next* function (incrementing a pointer) is broken down into two stages: (1) lifting up a foot; and (2) putting that foot down in the 'next' place. A stochastic process controls where the feet are put down, allowing them to be put down in the "wrong" place and so causing mutations in the copied string (variations in the copying process). The *next* function is embodied because the stochastic process depends on the composition of the walker chemical. Changing the walker chemical changes the stochastic process, and so an evolving walker chemical can change the way in which it performs its *next* function.

We show that this embodiment allows the walker chemical to change its mutation rate through evolution. This demonstrates the usefulness of an embodied *next* function (increment operation) used to make an embodied copy operation.



[	begin	Begin defining a binding site
]	end	End definition of a binding site
<	show	Show binding site
>	hide	Hide binding site
!	stop	Stop the execution of a program
a-z, 0-9	junk	Non-functional atoms

Figure 1: The alphabet of Graphmol atoms.

## Definition of Graphmol

The chemicals in Graphmol are represented by graphs. Each graph is both a data structure and a program. The execution of the program changes the structure of the graph.

A Graphmol chemical is defined by a string of atoms over an alphabet (figure 1). This string is parsed into three types of nodes (binding sites, which can be *shown* or *hidden*; functions; and junk), and folded into a graph with three types of edge (program edges, fold edges, and bind edges). Distances through the graph are used in a stochastic binding process (distances are calculated using the number of atoms in each node).

Reactions are defined by the Graphmol programming language, which has two parts: a declarative part (binding process) and an imperative part (instruction pointers).

The declarative part defines how chemical graphs bind to each other, implemented by a simple aspatial physics engine. This continually changes the graph structure by adding bind edges between *shown* binding sites. The process is stochastic, and the chance of two binding sites binding (having a bind edge added) depends on: (1) how closely their binding site patterns match; and (2) their distance apart, through the graph (measured as the length of the shortest path between the two binding sites).

The function nodes in the graph are the imperative language instructions. Instruction pointers move through the graph, executing the function nodes. This changes the graph structure by *showing* and *hiding* binding sites. When binding sites are *shown*, new binds become possible; when binding sites are *hidden*, some binds become impossible.

Junk affects how programs run in two ways: (1) it acts as a no-op for instruction pointers moving through the graph, slowing down execution of programs with respect to the timescale of the binding process; (2) it affects the graph distance between nodes, used to calculate binding probabilities.

**Parsing and Folding** There are two steps in converting a string of atoms into a chemical graph. These are (1) parsing atoms into nodes and (2) folding: connecting function nodes to their binding site nodes (figure 2).

A sequence of non-functional atoms enclosed in brackets [nnnnn] (with no internal brackets) defines a *binding site*. So the string [hdfgdd[icsd]bdgd[dhdhd]ixr]ss defines two binding sites, *icsd* and *dhdhd*. The string of

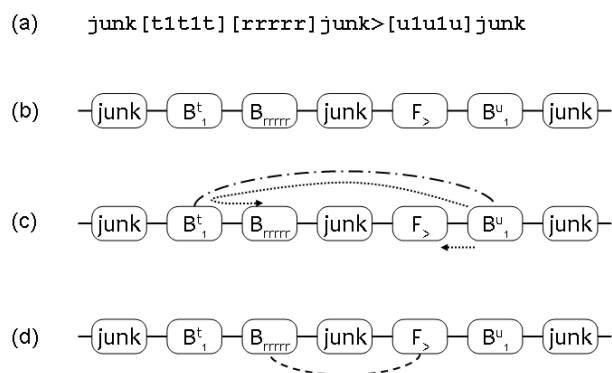


Figure 2: Parsing and folding: (a) a string of atoms; (b) the parsed graph of nodes connected by program arcs (solid edges); (c) temporary edge between *u1u1u* binding node and *t1t1t* binding node (dash-dotted edge), used to find closest functional node and binding site (dotted arrows); (d) resulting fold edge between the function node and its binding site (dashed edge).

atoms is parsed into a linear graph (figure 2 (b)) of binding site nodes, function nodes, and junk (everything else).

When executed, each < or > function *shows* or *hides* a particular binding site. The folding process connects these functions to the sites they affect. A temporary graph edge is added between a binding site node of the form *uxuxu* and its matching *txtxt* binding site node (where *x* is any non-functional atom). The closest *show* or *hide* function node, and closest the binding site node (measured along edges including the temporary edge), to the *uxuxu* node, are joined by a fold edge, and the temporary edge is removed. The result is the folded chemical graph (figure 2 (d)).

The fold edge is a form of indirect addressing. Instead of the function node specifying explicitly which binding site it *shows* (or *hides*), it instead specifies a template: *uxuxu*. During folding, this template is ‘dereferenced’ to locate the binding site: the closest binding site to the matching *txtxt*. Indirect addressing makes the system more evolvable, because the templates can change independently of the pattern of the target binding site.

**Chemicals** Once strings have been parsed and folded into chemical graphs, the graphs can start to react. The physics engine starts binding matching sites (here, we use exact string matching, so two binding sites either match or they do not; binding is a crisp process). When more than two sites match, the choice of which to bind is made stochastically, based on the graph distances between the sites.

When two binding sites bind, a bind edge is created between them, and an instruction pointer is created at each binding site. These instruction pointers move along their respective chemicals, executing any functions they reach.

As Graphmol runs, the graph states change because of two

processes: (1) instruction pointers move along the chemicals, executing functions that `show` and `hide` binding sites; (2) the physics engine makes binds happen between binding sites that match and are close together, which creates new instruction pointers.

**Reactions** There are two different concepts of ‘reaction’ in Graphmol: (1) a micro-scale interaction between two binding sites; (2) a macro-scale interaction between two chemicals, either designed into the chemicals, or an emergent property of the system.

(1) In the micro-scale case, a ‘reaction’ is the same as a bind between two binding sites. If two binding sites have patterns that match, then they have a probability of binding that depends on their distance apart through the graph. If the two binding sites are on different chemicals (that are not bound), then their distance apart is not defined, and they have a (pre-specified) low probability of binding.

When a bind happens, a bind edge is created between the two binding site nodes. This changes the topology of the chemical graphs, changing the probabilities of other binds happening. This new edge remains in place until one of its binding site nodes is `hidden`, at which point the edge is removed. When the bind happens, two instruction pointers are created, one at each binding site node. They move along their respective chemical’s program edges, executing any function nodes they encounter, until they reach either the end of the chemical, or a `stop`, (!), function, at which point the instruction pointer is removed.

The immediate result of this type of reaction is a graph topology change. The two chemicals are now connected together, and so the distances between binding sites have changed. A longer-term result of this reaction is that two computer programs are now running, represented by the two instruction pointers that are created. If another bind happens before these programs finish running, then further programs start executing in parallel.

This definition of ‘reaction’ views Graphmol as a simulation of nodes in a graph. Graphmol simulates these nodes by continually iterating the instruction pointers that exist (running the programs), and checking if any new binds happen (starting new programs). As the programs run, new binding sites become visible and so new binds can happen.

(2) In the macro-scale case, a ‘reaction’ is not defined explicitly as part of the Graphmol program: instead, it is a property of a running system. This can be an emergent property, produced by an evolutionary system. But in order to bootstrap evolutionary systems, we can design macro-scale reactions by hand-crafting Graphmol chemicals.

In traditional AChems, a reaction is a process whereby two chemicals are chosen to enter a black box, something happens, then one or more chemicals emerge from the box. Viewing Graphmol as a simulation of graph nodes does not fit this black box definition of a reaction. But we can use the

```
[start] junk
[lllll] ! junk [xxxxx] ! junk [rrrrr] ! junk
[lllll] ! junk [xxxxx] ! junk [rrrrr] ! junk
...
[lllll] ! junk [xxxxx] ! junk [rrrrr] ! junk
[lllll] ! junk [xxxxx] ! junk [rrrrr] ! junk
[stop]
```

Figure 3: The Graphmol DNA as a string of atoms (white-space added for readability only). The `xxxxx` binding sites are the bases that carry the information. The DNA chemical can be of arbitrary length.

simulation to implement *white* box reactions instead.

We can design two chemicals that have binding sites with matching patterns. We can set up the internal states of these chemicals so that only the two matching binding sites are `shown` (the rest being `hidden`). When we put these chemicals into the simulation, they will bind and start executing their programs. The execution of their programs might cause other binding sites to become `shown` and other binds to happen, but eventually all the programs will stop and no more binds will be possible. The individual programs cannot go into an infinite loop, since they execute along the program edges of a linear graph. The whole simulation could go into an infinite loop, but we assume not, for this argument.

We can think of this whole process as one ‘reaction’, and the system now looks like a traditional AChem, but with a complicated reaction mechanism. The chemicals that now exist in the simulation are the products of the ‘reaction’. Macro-reactions of this type are white boxes, because they are embodied in the simulation. This means that other chemicals can interfere with the process of the reaction.

## Embodied copying in Graphmol

Binding and program execution change the topology of chemical graphs. We use this to make one chemical graph move, relative to another. We make a long linear chemical graph composed of binding sites separated by regions of junk. This chemical contains no function atoms, so will not change its own topology. We make a second, smaller, chemical that ‘walks’ along the long chemical by alternately `showing` and `hiding` its six binding site ‘feet’. We add a special `crisp char-copy` instruction to the Graphmol language, specifically for the purpose of the experiments reported here.

The idea of a small chemical moving along a long, linear chemical is analogous to the way in which DNA is copied in biology. DNA is a long linear chemical. The chemical ‘DNA polymerase’ moves along the DNA and copies it. The actual process in biology is much more complicated than this, but making a simplified abstraction of the process allows us to implement an embodied copy operation in an AChem. Furthermore, many chemicals in biology move along DNA or RNA chemicals (not just to copy them). For example: (see

```

[t1t1t] [yyyyy] junk >[u4u4u] junk <[u2u2u] !
[t2t2t] [magic] junk >[u5u5u] junk <[u3u3u] !
[t3t3t] [eeeeee] junk >[u6u6u] junk <[u4u4u] !
[t4t4t] [yyyyy] junk >[u1u1u] junk <[u5u5u] !
[t5t5t] [magic] junk >[u2u2u] junk <[u6u6u] !
[t6t6t] [eeeeee] junk >[u3u3u] junk <[u1u1u] !

[tstst] [fgneg] junk <[u1u1u] junk >[ususu] !

[tetet] [fgbc] junk junk junk junk
junk >[u1u1u] junk >[u2u2u] junk >[u3u3u]
junk >[u4u4u] junk >[u5u5u] junk >[u6u6u]
junk <[ususu] junk >[ueueu] !

```

Figure 4: The Graphmol walker as a string of atoms. There are six feet (t1t1t–t6t6t), a ‘start’ site (tstst) and a ‘stop’ site (tetet). The length of the junk sections is varied in the experiment (see later).

any biology textbook for details, for example [1]) helicases (that unwind the two strands of DNA), ligases (that glue together sections of DNA) and ribosomes (that transcribe RNA into protein).

So, if we are interested in simulating analogies of biology, then movement of one chemical along another is a useful type of process to have in general.

**Graphmol DNA** We design a Graphmol chemical analogous to biological DNA. DNA stores information as a sequence of DNA bases attached to a common “backbone” structure.

Graphmol DNA has a sequence of ‘base’ nodes containing different information, interspersed with backbone nodes (figure 3). A ‘base’ node is a binding site, whose pattern is five information-carrying atoms (shown generically as xxxxx). Two backbone nodes [l1l1l1] and [r1r1r1] give the DNA a direction. (The stop atoms, !, are for efficiency, to remove the instruction pointer that is created on the DNA when a bind occurs.)

The junk regions add distance between the binding sites, which controls the probability of binding to different sites. In the implementation reported here, the DNA’s junk regions are each 40 atoms long.

The DNA chemical also has a start and a stop binding site. These allow the walker chemical to begin copying from the start of the DNA and to unbind when it reaches the end. This allows us to program the copy operation as a ‘macro-scale reaction’, as described above.

**Graphmol Walker** The walker chemical is shown in figure 4 as a sequence of atoms; its walking behaviour is shown schematically in figure 5. The walker chemical moves along the DNA chemical using six ‘feet’ (binding sites) alternating their visibility in a cycle. Feet 1 and 4 bind to [l1l1l1] on the DNA, feet 2 and 5 to [xxxxx], and feet 3 and 6 to [r1r1r1]. In this paper, binds happen if sites match exactly, where alphabet atoms match their complements (rotated 13 characters through the alphabet), and digits do not match.

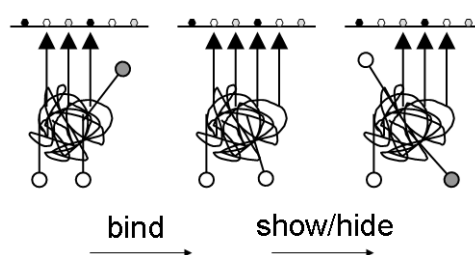


Figure 5: The walking process. Feet 1–3 are shown and bound (triangles), foot 4 is shown and unbound (dark circle), feet 5 and 6 are hidden (white circles). **bind**: The physics engine binds foot 4 (which is now shown with a triangle). **show/hide**: The bind starts a program running, which hides foot 1 (which therefore unbinds), and shows foot 5 (which is unbound). The cyclic process is ready to start anew.

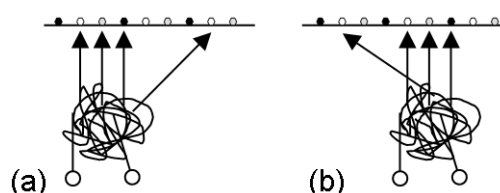


Figure 6: Low probability mis-stepping: (a) stepping over a site; (b) stepping backwards.

For the purposes of this paper, the Graphmol language is extended with magic binding sites that match and bind to any of the DNA’s [xxxxx] information-carrying binding sites. It performs a crisp copy of the bound node (a crisp char-copy function, from algorithm 1).

The walker has a ‘start’ [fgneg] region and a ‘stop’ [fgbc] region. The start region sets up the walker’s feet ready to begin moving along the DNA. The end region unbinds the walker from the DNA and sets the walker up ready to start another copy. This is a crisp start and at-end function, from algorithm 1.

Each foot has a short program associated with it. These programs show and hide the walker’s six feet in a cyclic pattern, making it walk along the DNA (figure 5).

Each of the walker’s feet has a pattern that matches multiple binding sites on the DNA. Because the probability of binding depends stochastically on graph distance, the walker’s feet will always be more likely to bind to sites on the DNA that are close to where the walker is currently bound. As three of the walker’s feet are always bound at the same time, the next matching binding site along the DNA will always be closer to the walker’s shown foot than earlier or later DNA sites. The walker usually steps to the correct next binding site, but can sometimes (with a low probability controlled by the amount of junk) jump forwards or step backwards (figure 6). Thus the walker implements an embodied next function (from algorithm 1).

Because the walker is copying the chemical it walks over. These jumps forwards and backwards correspond to insertions and deletions in the copied chemical.

Through the same binding process, the walker can also occasionally get its feet tangled, and fall off, resulting in a truncated copy. So the walker also implements an embodied `at-end` function. It has two `at-end` functions: a crisp one ('stop' region) and an embodied one (fall off early).

## Experiment

The Graphmol walker chemical, described in the previous section, can copy a DNA chemical, making insertion, deletion and truncation errors. But the way in which it makes these errors is not a collection of arbitrary choices written into an equation or a piece of stochastic code. It is a collection of arbitrary choices written into a machine (chemical) implemented in a lower-level stochastic programming language (Graphmol). If this machine/language combination is evolvable, then these arbitrary choices can be changed by evolution, and adapted to the problem being solved.

This paper is a feasibility study, testing that the embodied copying process implemented by the walker is evolvable. We show that, due to the design of the walker and of Graphmol, there is evolutionary pressure for the walker to evolve. It can trade off its accuracy against its speed of copying, by altering its level of junk. With more junk, the walker copies more accurately but also more slowly. With less junk, the walker copies less accurately but also more quickly.

### Experiment design

We want to test the hypothesis that changing the walker's junk level changes its speed and accuracy of copying.

To test this hypothesis, we run multiple simulations of the walker copying the DNA chemical. The length of the DNA chemical (number of bases) is the same as the length of the walker (number of atoms). This simulates the fact that if the walker was evolving, then changing its junk level would change the length of its encoding on the DNA.

We set up the DNA chemical by showing all of its binding sites. We set up the walker chemical by hiding all of its binding sites except the 'start' site [`fgneg`]. We then bind the walker's 'start' site to the DNA's 'start' site and simulate the (macro-scale) reaction until the walker unbinds from the DNA, thus finishing its copy. When the walker unbinds from the DNA, we compare its copy to the original DNA. The pattern of bases on the original DNA is randomly generated each time.

We repeat this copying process for walker chemicals containing different levels of junk. The junk regions in the walker chemical (see figure 4) are varied in length from one atom to 20 atoms. In this experiment, all of the junk regions within the walker are the same length as each other, for simplicity. If the walker was evolving, it would not need to enforce this. Indeed, unless there was evolutionary pressure

for it, evolution would probably not maintain 26 different regions at the same length. So this experiment shows a coarse view of the evolutionary options the walker has. In reality, the walker has a much finer level of control over its junk regions than this experiment shows.

For each different level of junk, we measure the time taken for the walker to make a copy (figure 7(a)) and the accuracy of its copying (figure 7(b)). Since the walker can make insertions, deletions and truncations of the DNA it is copying, there are many ways to define accuracy. We use the following. We care about the walker copying the DNA almost perfectly: we want perfect copies most of the time, but occasionally we want small mutations for evolution to exploit. So we define an 'almost perfect copy' as a copy that differs from the original by at most three bases, i.e. any combination of three insertions or deletions. To determine if a copy is almost perfect, we use Smith-Waterman alignment [5]. The Smith-Waterman algorithm measures the length of the longest common subsequence between two strings, taking into account (and penalising) short insertions and deletions. We set the penalty for an insertion or deletion to be 1, to measure the number of errors in the copy (subtracting the length of the original DNA, and taking the absolute value). If the number of errors is three or less, the copy is 'almost perfect'. Values other than three give qualitatively similar results, but larger values are more noisy so more experiments would need to be run to obtain the same error bars.

We run 80 copies per junk level, counting the number of nearly perfect copies to measure accuracy. We then repeat this process 20 times, to determine the error in these measurements (shown as notched boxplots in figure 7).

## Results

As the junk level increases, the walker takes longer to copy its DNA (figure 7(a)). This is for two reasons: (1) more junk makes the graph distance between binding sites longer, so the probability of the walker binding (and hence taking a step) is reduced; (2) more junk means the walker's encoding on the DNA is longer, so takes more steps to copy.

As the junk level increases, the walker becomes more accurate at copying its DNA (figure 7(b)). This is in spite of there being more DNA to copy at higher junk levels. As junk increases, the probability of binding is reduced in such a way that the probability of an erroneous bind (either jumping forwards or stepping backwards, figure 6) is reduced more than the probability of it making a correct bind (the probability is a non-linear function of distance,  $p(d) = (20/d)^{77}$ , chosen to give good behaviours over a range of chemical sizes). This makes the walker more accurate with more junk.

A walker with a low junk level is fast but error-prone; a walker with a high junk level is slow but reliable. So, the walker can trade off accuracy against speed. We can see this tradeoff by graphing the rate of copying for each junk level (figure 7(c)). This is the number of nearly perfect copies

made, divided by the time taken to make them. The graph is noisy at low junk levels because few nearly perfect copies are made here (as can be seen from the accuracy graph, figure 7(b)). The tradeoff can be seen in this graph as a peak at a moderate amount of junk. Too much junk and the walker copies too slowly, making its *rate* of accurate copying low. Too little junk and the walker makes too many errors, making its rate of *accurate* copying low.

## Discussion

When the walker is put into a simulation where it can evolve, it will be able to control its own junk level through mutations that add or remove junk. These results show that changing the walker's junk level changes its speed/accuracy tradeoff for copying. Thus the walker will be able to find, for itself, the tradeoff between speed and accuracy that optimises its survivability in its environment.

Because it finds this tradeoff for itself, it will be able to re-optimize if its environment changes. We have taken a quantity that is normally a parameter in ALife simulations, the mutation rate, and embodied the process that requires this parameter. This means that the ALife organisms can change this parameter, by manipulating the underlying processes that give rise to the parameter. The mutation rate has changed from being an external parameter, to an observed property of the system.

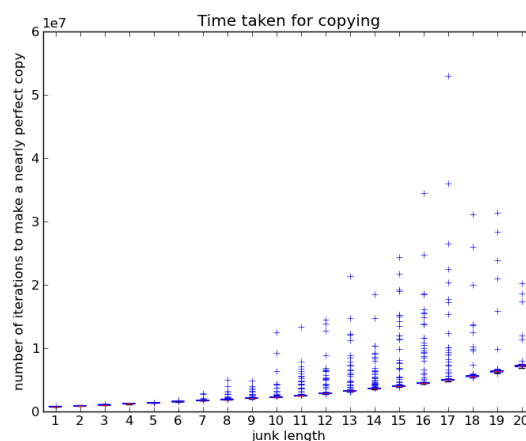
## Future work

This experiment has demonstrated that it is possible to build an AChem with an embodied copying process that can be exploited by the system to adapt its mutation rate. But because the whole *process* of mutation is embodied (not just the rate), the system should be able to change the copying process, generating novel *types* of mutation. When we run the embodied copying process in a evolutionary system, we will be looking for such changes.

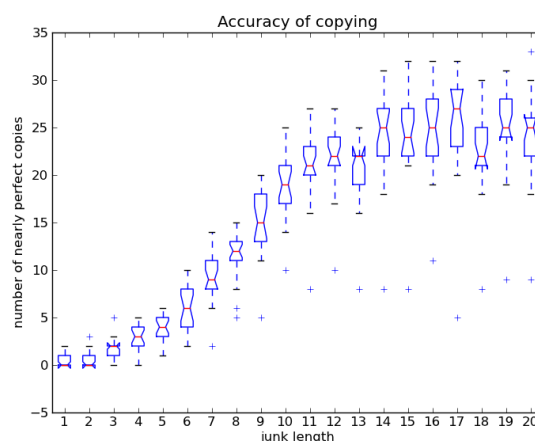
To make systems that can change their mutation process in different ways, different parts of the copying process can be embodied:

## Copying a character

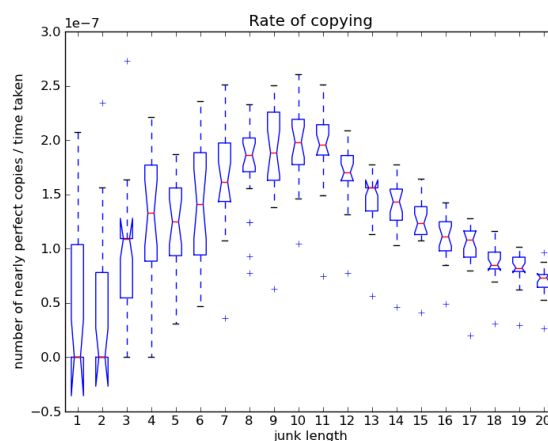
The walker chemical takes the process ‘iterate over a string’ and implements this as an embodied process in Graphmol. Here we have used a crisp `char-copy` function to copy each character of the string (so the only copying errors are insertions and deletions). The `char-copy` function could instead be made stochastic, to explore the effect of point mutations on the walker. More interestingly, the `char-copy` process could be embodied, by implementing a `char-copy` mechanism in Graphmol. Just as we broke the string copy process into components (algorithm 1), we can break the character copy process into components to be embodied (algorithm 2):



(a) Copying time increases with junk level (so speed decreases).



(b) Accuracy of copying increases with junk level.



(c) Rate of copying has an optimum junk level, trading off speed against accuracy.

Figure 7: How the walker's junk length affects its copying. A “nearly perfect copy” is a copy that differs from the original by at most three bases (three insertions or deletions). The notches show the 95% confidence intervals.

---

**Algorithm 2** Deconstructing the character copy operation, as a prerequisite to embodying it

---

**result** char\_A := char\_B

```
x := read(char_B)
y := repn(x)
char_A := write(y)
```

---

1. **Read the character** `read` This could be implemented as a set of binding sites with patterns that match each of the DNA bases. When one of these sites binds to a DNA base, a program on the walker changes the walker's state.
2. **Represent the character** `repn` The walker needs to know which site has been bound, so needs a change of state to signify this. For example, it could show a binding site corresponding to the base that it is currently copying.
3. **Write the character** `write` The walker needs to maintain a chemical representing the copy it is making of the DNA chemical. The binding site it shows in the step above, could bind to a free-floating chemical base, at which point the walker would attach this base to the copy.

After attaching the copied character to the result string, the walker needs to move on to the next character on the string being copied. The walker already does this to walk along a DNA chemical, but it will also need to do this with the copy it is producing.

### Making binding evolvable

In this paper, binding requires an exact (complementary) match between binding sites. We need to allow 'soft' binding, so that evolution can modulate binding affinities to give complex behaviours [3]. This will make the `start` and `at-end` functions embodied (and `char-copy`, if used with the previous section).

In this paper, the folding sites `[uxuxu]` and `[txtxt]` (figure 2a) were chosen arbitrarily. In future, the folding process will be implemented by a folding chemical, with binding sites matching `[uxuxu]` and `[txtxt]`. This makes the folding process embodied, rather than hardcoding folding in the definition of Graphmol. Thus evolution will be able to exploit the folding process and potentially change it.

### Conclusions

We have discussed how ALife simulations can be made more evolvable by making their copying process embodied rather than stochastic. An existing example of where an embodied copying operation has led to interesting behaviour is the Stringmol AChem [3].

We have embodied copying in a new way, by making an embodied `next` function (increment operation). This involved designing the Graphmol AChem and implementing

an embodied `next` function in Graphmol. We attached a crisp `char-copy` function to this embodied `next` function, creating an embodied string copy operation. This embodied string copy operation can make insertion and deletion mutations on the copied string.

We have run a feasibility experiment (figure 7) to show that the embodied copy operation is evolvable, and has the potential to adapt its own mutation rate to its environment. But more experiments are needed to find the environments in which it will show this. The copying process adapts by changing the level of junk in its embodiment, which changes its probability of incrementing correctly versus incrementing erroneously. In this way, the embodied copy operation can adapt its own mutation rate without there being an explicit mutation rate parameter in the system.

This is the crucial difference between embodied systems and stochastic systems. Stochastic systems are crisp systems with parametrised variation added in. Embodied systems are evolvable machines that can evolve their own parameters and processes, because they are implemented in a lower-level language.

### Acknowledgements

This work is part of the Plazmid project, EPSRC grant EP/F031033/1. We thank the other members of the Plazmid project for valuable comments and ideas: Ed Clark, Simon Hickinbotham, Peter Young, Tim Clarke and Mungo Pay. And the anonymous referees for their helpful comments.

### References

- [1] T. Brown. *Genomes 3*. Garland Science, 2006.
- [2] S. Hickinbotham, E. Clark, S. Stepney, T. Clarke, A. Nellis, M. Pay, and P. Young. Molecular microprograms. In *ECAL 2009, Budapest, Hungary, September 2009*. LNCS. Springer, Sept. 2009.
- [3] S. Hickinbotham, E. Clark, S. Stepney, T. Clarke, A. Nellis, M. Pay, and P. Young. Diversity from a monoculture: Effects of mutation-on-copy in a string-based artificial chemistry. In *ALife XII*, pages 24–31. MIT Press, 2010.
- [4] S. Hickinbotham, E. Clark, S. Stepney, T. Clarke, A. Nellis, M. Pay, and P. Young. Specification of the stringmol chemical programming language version 0.1. Technical Report YCS-2010-457, Dept Computer Science, University of York, 2010.
- [5] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.
- [6] S. Stepney. Embodiment. In D. Flower and J. Timmis, editors, *In Silico Immunology*, chapter 12, pages 265–288. Springer, 2007.

# An Experiment in Mixing Evolving and Preprogrammed Robots

Sancho Oliveira<sup>1</sup>, Luis Nunes<sup>1</sup> and Anders Lyhne Christensen<sup>1</sup>

<sup>1</sup>Instituto de Telecomunicações & Instituto Universitário de Lisboa (ISCTE-IUL), Lisboa, Portugal  
sancho.oliveira@iscte.pt

## Abstract

Artificial evolution of robot behavior is commonly conducted in environments containing a single robot or multiple robots that are all controlled by evolving behavioral logic. In this paper, we take a novel approach and study how the presence of preprogrammed robots affects the evolutionary process and the solutions evolved. We evolve behavioral control that enables robots to forage. The robots are situated in an environment that contains a nest and a number of prey. The robots must either push or carry the prey to the nest. We analyze the behaviors evolved in mixed setups in which one or more preprogrammed robots are present. We compare these behaviors to behaviors evolved in non-mixed setup in which no preprogrammed robots are present. The results show that although the evolved robots do not use their capacity to communicate, they do collaborate with the preprogrammed robots. We find that the performance of some of the solutions evolved in the mixed setup is higher than the performance of homogeneous groups of robots.

## Introduction

In this paper, we take a novel approach to the evolution of behavioral control for robots. We report on experiments in which we evolve behaviors for robots that share the environment with preprogrammed robots. The preprogrammed robots are (aside from their behavior) indistinguishable from the evolving robots. Mixing evolving robots with preprogrammed robots is interesting for several reasons: from an engineering perspective, artificial evolution may be used to fill in the gaps between partially known (easily preprogrammable) solutions to complex tasks and/or to optimize the performance of a robot collective. From an evolutionary perspective, it is interesting to evaluate how the presence of robots programmed with a solution influences the evolutionary process and the solutions evolved – such as determining whether the evolving robots adopt the preprogrammed solution and/or whether they learn to communicate with the preprogrammed robots.

We use a multirobot foraging task for our experiments. A robot can push prey or it can pick up and carry a prey. If a prey-carrying robot collides with another robot, it loses

the prey. Thus, the robots must avoid collisions when carrying prey. The preprogrammed robots have the same sensory and actuation capabilities as the evolving robots. Each robot can control the color of its body. Whenever carrying prey, a preprogrammed robot sets its body color to red. When not carrying a prey, a preprogrammed robot sets its body color to green. Thus, nearby robots can see when a preprogrammed robot is carrying a prey or not and give way in order to avoid collisions. Since evolving robots have control over their body color too, they have the potential to communicate to nearby teammates in the same way as the preprogrammed robots do.

In this study, we analyze and discuss the fitness trajectories and the solutions obtained in evolutionary runs where one preprogrammed robot and two evolving robots are present. We discuss if and how the robots collaborate and communicate. We setup an experiment in which we take an incremental approach to evolution in order to increase the rate of solutions with a high average fitness. Finally, we report on experiments in which three preprogrammed robots and six evolving robots are present during evolution.

The contribution of this paper is three-fold: i) We demonstrate that evolving robots can learn to collaborate with preprogrammed robots. ii) We demonstrate how a basic incremental approach to evolution can increase the rate at which collaborative solutions are evolved when preprogrammed robots are present. iii) We show that heterogeneous groups of preprogrammed robots and evolved robots can achieve a better performance than homogeneous groups of preprogrammed robots.

## Related work

Interest in evolutionary robotics started in the early 90s (Cliff et al., 1993; Nolfi and Floreano, 2000). Initially, focus was on evolving a controller for a single robot to perform relatively simple tasks such as obstacle avoidance, exploration, and navigation (see for instance Nolfi et al. (1994)). Recently, there have been several studies on the evolution of controllers for multirobot systems—particularly those systems in which control is decentralized



and in which individual robots have limited sensory capabilities. In swarm robotics research (Şahin, 2005), it has been demonstrated how the application of evolutionary robotics can overcome the fundamental design problem of deriving microscopic rules for the individuals such that the desired macroscopic behavior emerges. When artificial evolution is applied to swarms of robots, the designer can specify a fitness function that scores the collective behavior and let evolution search the space of individual behaviors. Using this approach, Dorigo et al. (2004) demonstrated how a group of homogeneous robots could be evolved to aggregate and to display coordinated-motion when physically connected to each other. In another study, Trianni et al. (2006) demonstrated how a group of evolved homogeneous robots could cooperatively avoid holes.

Evolutionary robotics has been applied to heterogeneous multirobot systems: Tuci et al. (2008) evolved homogeneous controllers for heterogeneous robots. Nolfi and Floreano (1998) co-evolved a predator agent and a prey agent. The fitnesses of the two types of agents were co-dependent although each had a different genome.

It has also been demonstrated that heterogeneity can arise in a homogeneous system (identical agents with identical neuro-controllers). Quinn et al. (2003) evolved controllers for a team of three robots with minimal sensory capabilities. The robots' task was to aggregate and then travel a distance of one meter as a group. Interestingly, the team members dynamically adopted roles and moved in a line formation. The robot that would adopt the role as the leader, moved backward in order to perceive the middle robot. The middle and rear robot, on the other hand, moved forward. Ampatzis et al. (2009) evolved homogeneous controllers for two real robots that allowed them to *self-assemble*, that is, physically connect to one another. However, the robots first had to allocate roles so that one would be the gripping robot, while the other would be the gripped robot. The roles were allocated during what can be described as a dance: the robots would circle each other while performing oscillatory movements until one would approach the other to perform the grip.

In this study, we use a novel evolutionary setup. We explore the effect of the presence of preprogrammed robots on the evolved behaviors. We find that the heterogeneity in the group composition leads to role allocation and collaboration.

## Robot Model and The Task

Below, we start by presenting the robot model that we use. We go on to describe the foraging task and the environment. Finally, we briefly discuss the software simulator in which we conduct our experiments.

### The Robot Model

We use a differential drive, cylindrical robot model. Each robot has a diameter of 10 cm. The set of actuators is composed of two wheels, a prey carry mechanism and a change-

able body color. The two wheels can be controlled independently allowing a robot to move and to turn. Gaussian noise with standard deviation of 5% is added independently to the left wheel speed and to the right wheel speed set by the robot controller in order to simulate issues such as slippage, slightly uneven ground and so forth. The prey carry mechanism enables a robot to pick up a prey within a distance of 5 cm. The body color actuator has three possible settings: green, red, and black. Whenever green or red, a robot can be detected by other nearby robots, while when black, the robot is invisible to other robots.

The robots are equipped with several sensors that allow them to perceive i) whether they are currently carrying a prey or not (prey-carried sensor), ii) whether they are inside the nest or not (in-nest sensor), and iii) the presence of nearby objects: eight nest sensors, eight prey sensors, eight red robot sensors, and eight green robot sensors.

Aside from the prey-carried sensor and the in-nest sensor, all the sensors operate in a similar way, but register different types of objects. The nest sensors only register the nest. The prey sensors only register prey. The green robot sensors only register green robots. The red robot sensors only register red robots. The sensors are distributed evenly around the robot's body.

A sensor only registers objects within a certain distance and angle with respect to its orientation. All sensors have an opening angle of  $135^\circ$  and a range of 1 meter, except for the nest sensors which have a range of 10 meters. If there are no sources within sensor's range and opening angle, its reading is 0. Otherwise, the reading is based on distance to the closest source ( $c$ ) according to the following equation:

$$s = \frac{\text{range} - d_c}{\text{range}}, \quad (1)$$

where  $\text{range}$  is the sensor's detection range and  $d_c$  is the distance between the closest source  $c$  and the sensor.

### The Foraging Task

Our experiments are conducted in the arena shown in Figure 1. The robots must search for prey and transport them to the circular nest area with a diameter of 0.50 m centered in the arena. The nest can be perceived by the robots using their nest sensors. The prey are scattered in the foraging area around the nest. The foraging area is circular and has a diameter of 4 meters. Whenever a prey is dropped in the nest, it is immediately redeployed to a random location in the foraging area. 13 prey are present in the environment which results in a prey density of 1 prey/m<sup>2</sup>. When a prey-carrying robot collides with another robot, it loses the prey that it was carrying. The lost prey is randomly redeployed in the environment in order to keep the prey density constant.

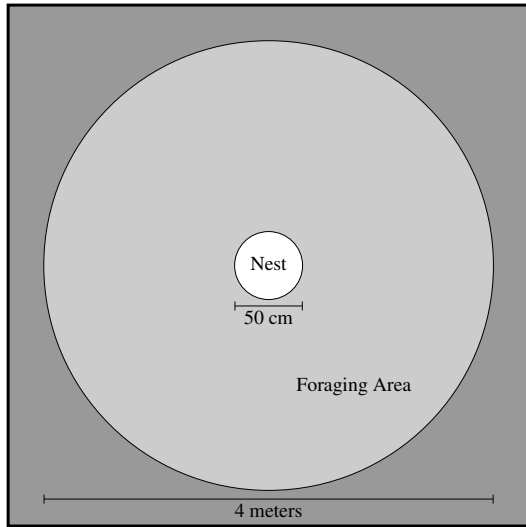


Figure 1: Foraging arena.

## Simulation Environment

We have implemented the robot model and constructed the environment discussed above in JBotEvolver (see <http://sourceforge.net/projects/jbotevolver>). We have implemented our own neuro-evolution framework that allows for distributed, fault tolerant fitness evaluation.

## Controller Architecture

Below, we present the control logic for the preprogrammed robot and the artificial neural network used for the evolving robots.

### Preprogrammed Robots

A finite state machine representation of the control program for the preprogrammed robots is shown in Figure 2. A preprogrammed robot starts of in the “Search” state in which it locates and moves towards the nearest prey. If the preprogrammed robot detects the presence of a red robot in its way, it assumes that the red robot is carrying a prey and therefore turns around (180°) and moves out of the way (state “Make way”). When a prey is encountered, the preprogrammed robot attempts to pick it up (state “Pick up”). If the prey is picked up successfully, the preprogrammed robot becomes red and starts moving towards the nest (state “Transport”). When the nest is reached, the preprogrammed robot drops the prey (state “Drop”) and returns to the “Search” state. If the preprogrammed fails to pick up a prey or if it loses the prey (due to a collision), the preprogrammed robot returns to the “Search” state.

In the finite state machine in Figure 2, we have colored the states with the color that a preprogrammed robot has when in the respective states. Whenever a prey is carried, the preprogrammed robot is red. Otherwise, the preprogrammed

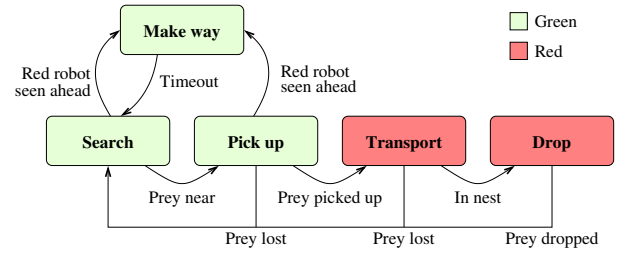


Figure 2: Preprogrammed controller.

robot is green. Preprogrammed robots are never black.

### The Evolving Robots

The evolving robots are controlled by a continuous time recurrent neural network (Beer and Gallagher, 1992). The network consists of three layers of neurons: an input layer with 34 neurons, a hidden layer with 5 neurons, and an output layer with 4 neurons. The input neurons  $I_i$  are reactive. The prey-inputs ( $I_1$  to  $I_8$ ), the nest-inputs ( $I_9$  to  $I_{16}$ ), the green-inputs ( $I_{17}$  to  $I_{24}$ ), and the red-inputs ( $I_{25}$  to  $I_{32}$ ) are all set based on sensor readings from the respective sensors. The prey-carried-input ( $I_{33}$ ) is 1 if a prey is currently carried and 0 otherwise. The in-nest-input ( $I_{34}$ ) is 1 if the robot is in the nest and 0 otherwise. The neurons in the hidden layer are fully connected and governed by the following equation:

$$\tau_i \frac{dH_i}{dt} = -H_i + \sum_{j=1}^{34} \omega_{ji} I_j + \sum_{k=1}^5 \omega_{ki} Z(H_k + \beta_k), \quad (2)$$

where  $\tau_i$  is the decay constant,  $H_i$  is the neuron's state,  $\omega_{ji}$  the strength of the synaptic connection from neuron  $j$  to neuron  $i$ ,  $\beta$  the bias terms, and  $Z(x) = (1 + e^{-x})^{-1}$  is the sigmoid function.  $\tau$ ,  $\beta$ , and  $\omega_{ji}$  are genetically controlled network parameters. The possible ranges of these parameters are:  $\tau \in [0.1, 32]$ ,  $\beta \in [-10, 10]$  and  $\omega_{ji} \in [-10, 10]$ . Cell potentials are set to 0 when the network is initialized and circuits are integrated using the forward Euler method with an integration step-size of 0.2.

The output layer is fully connected to the neurons in the hidden layer. The activation of the output neurons is given by the following equation:

$$O_i = \sum_{j=1}^4 \omega_{ji} Z(y_j + \beta_j); \quad (3)$$

The first two outputs  $O_1$  and  $O_2$  control the speed of the left and the right wheel, respectively. Their output is linearly mapped to speeds in the range  $[-50 \text{ cm/s}, 50 \text{ cm/s}]$ . The third output  $O_3$  is mapped to the prey carrying mechanism: if  $O_3 > 0.5$ , the robot attempts to pick up the closest prey or to hold a prey if one is already carried. If  $O_3 \leq 0.5$ , any prey

carried will be dropped. The fourth output  $O_4$ , controls the color of the robot. For values in the range  $[0, 0.33]$  the robot becomes invisible to other robots, for values in the range  $]0.33, 0.66[$ , the robot becomes green, and for values in the range  $[0.66, 1.00]$ , the robot becomes red.

## Evolutionary Algorithm

We use a simple generational evolutionary algorithm (Schwefel, 1995; Goldberg, 1989). Each generation consists of 100 genomes. Each genome consists of a vector of 228 real valued numbers. These values encode the weights of the synaptic connections between neurons, the bias terms and the decay constants for a neural network with the topology described in the previous section. After sampling the fitness of each genome in a generation, the 5 best genomes are retained and the rest are discarded. These 5 genomes are the parents of the subsequent generation. From each parent an equal number of children (19) are created and the parents are copied to the new generation. The genotype for a child is obtained adding a random Gaussian offset to each real-valued gene with a probability of 15%.

We compute the fitness at the group-level. Thus, in the experiments where a preprogrammed robot is present, its behavior and its performance contribute to the fitness of the group in the same way as the behavior of the evolving robots. The fitness function  $F(i)$  is given below:

$$F(i) = P_i + \sum_{s=1}^{\text{time-steps}} f_{i,s} \quad (4)$$

where  $i$  is the genome being evaluated,  $P_i$  is the number of prey foraged and  $f_{i,s}$  is computed at every time-step,  $s$ . The term  $f_{i,s}$  is computed in the following way:

$$f_{i,s} = 10^{-3}c_s + 10^{-4}d_s \quad (5)$$

where  $c_s$  is the number of robots carrying a prey at time-step  $s$  and  $d_s$  is a prey distance reward that depends on the distance between each prey and the nest at time-step  $s$ . The prey distance reward is computed using the formula:

$$d_s = \frac{1}{n^{\circ} \text{ prey}} \sum_{j=1}^{n^{\circ} \text{ prey}} \frac{1.75 \text{ m} - \text{dist}(p_j, \text{nest})}{1.75 \text{ m}}. \quad (6)$$

We sample the fitness of each genome five times and selection is based on the average fitness obtained.

## Results and Discussion

We initially experimented with two different evolutionary setups: a *mixed* setup in which two evolving robots and one preprogrammed robot were present, and a *non-mixed* setup in which three evolving robots were present. In each setup,

we performed 30 evolutionary runs with different initial random seeds for 2000 generations each. Each generation consisted of 100 genomes. The fitness of each genome was sampled in five trials of five minutes of virtual time (3000 control steps) each.

Below, we provide an overview of the results obtained. We then describe the different types of behaviors evolved in the *non-mixed* setup and the *mixed*, respectively. We go on to discuss cooperation and communication. We then experiment with incremental evolution in order to speed up evolutionary learning. Finally, we experiment with setups in which nine robots are present.

## Fitness Trajectories

The plot in Figure 3 summarizes the results of the evolutionary runs conducted in the *mixed* setup and in the *non-mixed* setup, respectively. The figure shows the average fitness of the best genome in each generation in all the 30 runs conducted in the *mixed* setup and in the *non-mixed* setup, respectively. We have included the fitness trajectory for the single highest scoring *mixed* run and for the single highest scoring *non-mixed* run. The horizontal line at  $y = 119.9$ , shows the average fitness obtained by three preprogrammed robots alone in the environment.

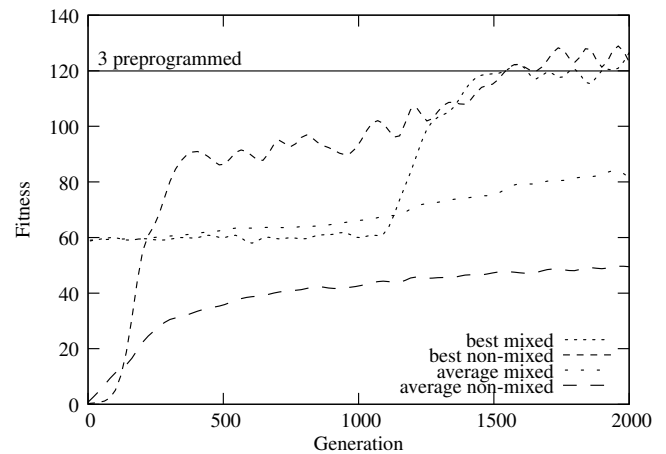


Figure 3: The fitness scores of the best and the average the best genomes in all the runs in the *mixed* setup and in the *non-mixed* setup. The horizontal line at  $y = 119.9$  indicates the average performance of a team of three preprogrammed robots.

The results in Figure 3 show that the fitness of the best genome in the *mixed* setup is on average higher than the best genomes in the *non-mixed* setup. The higher fitness in the beginning of an evolutionary run in the *mixed* setup is explained by the presence of the preprogrammed robot. The preprogrammed robot finds and transports prey to the nest from the onset of an evolutionary run whereas the evolving robots first have to learn to forage.

When a preprogrammed robot is alone in the environment, it obtains an average fitness of 60.0. When three preprogrammed robots are present in the environment, they interfere with one another. Furthermore, as a trial progresses, prey tend to be distributed further from the nest since more preprogrammed robots tend to forage the prey close to the nest faster. Interference and the increased prey distance both have negative impacts on the fitness score. Three preprogrammed robots therefore obtain a fitness (119.9) that is less than three times what a single preprogrammed robot obtains on average (60.0). In the beginning of an evolution run when the evolving robots are not yet foraging, the preprogrammed robot can often forage undisturbed in the *mixed* setup. The average fitness in the beginning of an evolutionary run in the *mixed* setup is therefore close to the fitness obtained by a single preprogrammed robot operating alone.

## Behavioral Analysis

In this section, we analyze the evolved robots' behaviors. A summary of the post evaluation scores for the 30 evolutionary runs conducted in the *non-mixed* setup and the 30 evolutionary runs conducted in the *mixed* setup can be seen in Figure 4. In the plot, we have grouped the evolutionary runs according to their foraging behaviors and fitness.

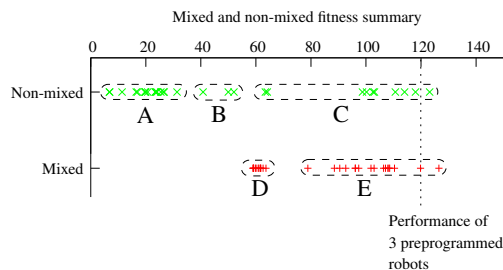


Figure 4: Summary of the post evaluation of the best behavior evolved in each evolutionary run in the *non-mixed* setup and in the *mixed* setup. We have divided the evolved solutions into groups A to E based on fitness and behavior.

In the *non-mixed* evolutionary runs, we observed behaviors that can be divided into three groups: A, B and C. All the solutions in all groups successfully forage prey, however, they forage in different ways. The behaviors group A all rely on pushing prey towards the nest. An example of the pushing behavior can be seen in Figure 5. The pushing behavior requires the robots to move in small circular patterns to constantly get behind the prey and the behavior is thus not very efficient.

The behaviors in group B rely on continually picking up and dropping prey. When a prey is picked up, it is often dropped after a single or a few control cycles, only to be picked up again immediately. One of the behaviors in group B is particularly interesting: often the robots transport two or

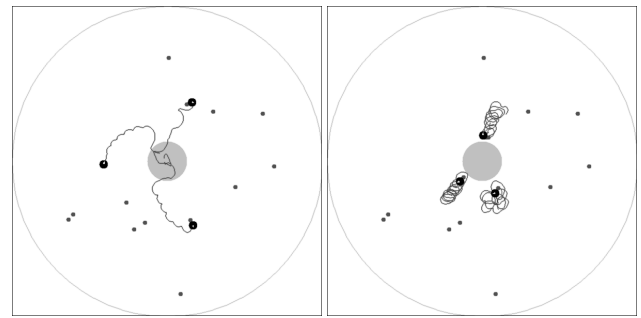


Figure 5: An example of a behavior in group A evolved in the *non-mixed* setup (two screenshots from the same experiment). The robots forage by pushing prey towards the nest. As can be seen on the figure, this behavior results in a lot of small circular movements and is thus not very effective.

more prey at a time by repeatedly picking up, dropping different prey. An example can be seen in Figure 6. Transporting multiple prey, however, comes at a cost: since a robot can only carry one prey at a time, it has to constantly make small circular movements to pick up the prey left behind. This means that the average fitness of the behavior in group B is lower than the average fitness of the behaviors in the last group of behaviors evolved in the *non-mixed* setup, group C.

In group C, the robots pickup prey and transport the prey back to the nest. The differences in fitness between the different solutions are due to a number of factors: how the robots search for prey, how efficient they are in moving to a prey once they have located the prey, and if and how much they interfere with one another. Some robots move away from the nest in a straight line to search for prey, some robots circle away from the nest, while in other cases, the robots move in more irregular patterns. Most of the robots move only forward or only backward, however, for some behaviors, the robots change direction once a prey is picked up. Changing direction is especially efficient for those robots that move directly from the nest to a prey: when a prey is picked up, they change direction (without having to turn around) to transport the prey back to the nest. Examples of some of the behaviors in group C can be seen in Figure 7.

We have divided the behaviors evolved in the *mixed* setup into two groups: D and E (see Figure 4). Group D contains the lowest scoring behaviors evolved in the *mixed* setup. The evolving robots in this group do not contribute to the foraging, but instead move away from the nest in order to let the preprogrammed robot forage undisturbed. In some cases, the evolving robots move beyond the foraging area, in some cases the evolved robots remain in the foraging area, and sometimes they even pickup prey.<sup>1</sup> However, in none of the cases do the evolved robots attempt to move prey closer to

<sup>1</sup>Carrying prey is rewarded in the fitness function (see the "Evolutionary Algorithm" section).

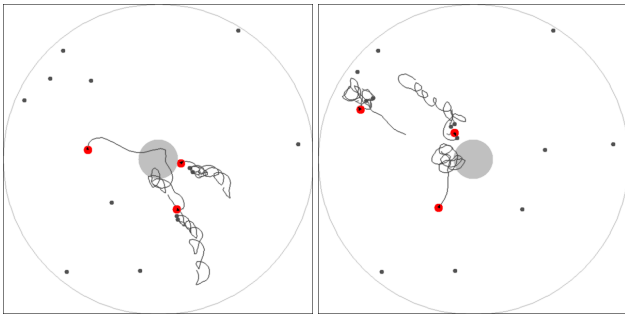


Figure 6: An example of the behavior in group B evolved in the *non-mixed* setup (two screenshots from the same experiment). By continually picking up and dropping prey, the robots are able to transport multiple prey towards the nest at the same time.

the nest.

The evolved solutions in group E all obtained an average post evaluation fitness of more than 70. In all of these solutions, the evolved robots actively forage. The difference in performance is due to the way in which the evolved robots search for prey: some of the evolved robots move directly towards prey close to the nest while others circle the foraging area and forage mainly prey located far away from the nest (thereby leaving the prey close to the nest for the preprogrammed robot to pickup). This type of behavior indicates that the evolved robots collaborate with the preprogrammed robot.

**Collaboration** To examine the level of collaboration (if any) between the evolving and preprogrammed robots, we analyzed if there is some evidence of division of labor: we recorded the number of prey foraged by evolved robots and the number of prey foraged by the preprogrammed robot in the *mixed* setup. We ran 100 trials with each of the highest scoring genomes from the 30 evolutions conducted in the *mixed* setup. For 16 of the 30 genomes, the preprogrammed robot forages significantly more prey when the evolved robots are present compared to when it is the only robot in the environment (Mann-Whitney,  $p < 0.05$ ).

When the preprogrammed robot is alone, it forages 57.9 prey on average during a five minute trial, while when three preprogrammed robots are present, each forages on average 38.7 prey. When evolved robots are present, the preprogrammed robot forages an average of 75.3 prey per trial for the best solution in the *mixed* setup. These results indicate that the evolved robots have learned to collaborate with the preprogrammed robot. For the best solution in the *mixed* setup, the average distance (over 100 five minute trials) of the preprogrammed robot from the center of the nest was 0.54 m, while the average distance of the each of the two evolved robots from the center of the nest was 1.06 m.

The evolving robots forage prey that are located far from the nest and leave the prey close to (but not always in) the nest. The preprogrammed robot (which prioritizes prey located close to the nest) then transports the prey left by the evolving robots the rest of the way to the nest. The division of labor is efficient because the evolved robots in general operate far from the nest, while the preprogrammed robot operates close to and in the nest – collisions are therefore avoided.

**Communication** The robots in both the *non-mixed* and the *mixed* setups have the capacity to change their body color and to detect the body color of nearby teammates. This capacity potentially allows the robots to communicate. However, in 22 out of 30 evolutionary runs in the *non-mixed* setup, the evolved robots remain mainly black (invisible to one another) during experimental trials. In the remaining 8 runs, the robots either remain mainly red (5) or constantly change color (3) during a trial.

In order to determine if communication plays a major role in the evolved solutions, we ran three sets of experiments in the *non-mixed* setup, where we fixed the body color of all the robots to black, red and green in 100 trials each. The differences in terms of performance when the body color is fixed and when the neural network has the control over the body color were minimal. The average performance difference was only 0.5%, with the largest drop being 3.7% and the largest increase in performance being 5.4%.

In a similar set of experiments in the *mixed* setup, we fixed the color of the preprogrammed robot and the two evolving robots. Fixing the body color to red results in an average performance drop of 22.6%. This drop is explained by the fact that the preprogrammed robot attempts to make way each time it encounters a red robot. The average difference in performance when the body color is fixed to either black or green and when the controllers have control over the body color was 0.6% with the largest difference being 3.2%. This indicates that the performance of the evolved robots does not depend on their capacity to change their body color.

It is surprising that the robots did not evolve to exploit their capacity to change color in the *mixed* setup to communicate with the preprogrammed robot (which already communicates its internal state by changing color depending on whether it is carrying a prey or not). A probable explanation for the lack of communication is that the robots can forage efficiently in the *mixed* setup without communicating. As discussed in the previous section, the evolved robots do in most cases learn to collaborate with preprogrammed robot by transporting prey located far from the nest closer to the nest for the preprogrammed transport the rest of the way to the nest. The robots operate in different regions of the environment and they do therefore not need to communicate in order coordinate their actions or to avoid collisions.

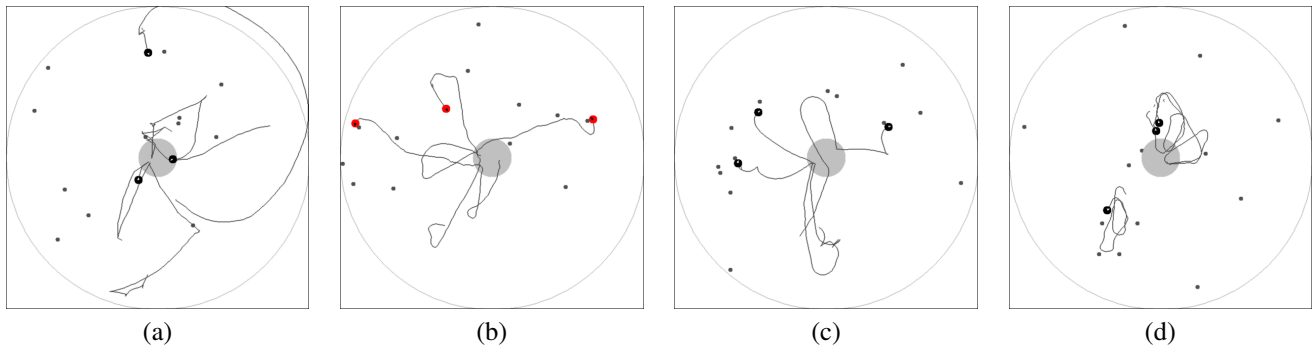


Figure 7: Examples of the behaviors from group C evolved in the *non-mixed* setup (screenshots from different experiments with different controllers). (a) the robots move directly to prey or start circling the foraging area in case no prey is found. (b) the robots have set their body color to red and turn around once a prey is picked up. (c) the robots move to prey in arcs. (d) robots pick up and carry prey, but they often interfere with one another.

**Incremental Evolution in the *mixed* setup** Of the 30 evolutionary runs conducted in the *mixed* setup, the 12 runs in group D did not evolve foraging behaviors, but instead, the evolved robots move away from the nest in order to avoid interfering with the preprogrammed robot. This solution is a local maximum in the fitness landscape because the preprogrammed robot is an efficient forager from the onset of the evolutionary process and any interference – a lost prey due to a collision for instance – would result in a lower collective fitness. We set up a series of experiment in which we tried to increase evolutionary pressure towards solutions in which the evolving robots participate in the foraging by initially reducing the speed of the preprogrammed robot. When the preprogrammed robot moves at a reduced speed, it forages less than when moving at full speed. Evolutionary pressure towards solutions in which the evolving robots actively forage is thus increased because any contribution made by the evolving robots proportionally is higher with respect to the fitness obtained by the team than when the preprogrammed robot is moving at full speed. In a new *incremental mixed* setup the preprogrammed robot initially moved at 50% of the full speed. Once a collective fitness of 50 was reached by the highest scoring individual in a generation, the speed of the preprogrammed robot was increased to full speed.

We performed 30 evolutionary runs in the *incremental mixed* setup. Out of the 30 evolutionary runs, only 6 produced non-foraging behaviors compared to 12 in the normal (non-incremental) *mixed* setup. The average of the post evaluation fitness of the best genome from each run in the *incremental mixed* was 90.0 compared to 84.7 in the *mixed* setup. For 24 of the 30 genomes, the preprogrammed robot forages significantly more prey when the evolved robots are present compared to when it is the only robot in the environment (Mann-Whitney,  $p < 0.05$ ). Hence, in the *incremental mixed* setup, the evolving robots learn more frequently to collaborate with the preprogrammed robot than in the non-

incremental *mixed* setup. Visual inspection of the successful solutions evolved in the *incremental mixed* setup confirmed that they are similar to the successful solutions evolved in the non-incremental *mixed* setup (that is, the behaviors in group E in Figure 4).

**Performance in larger *mixed* groups** In order to determine if and how the mixture of preprogrammed and evolved could benefit larger groups of robots, we conducted experiments in which nine robots were present in the environment: three preprogrammed robots and six evolving robots. We conducted the evolution in the same environment and with the same fitness function as used above. We used an incremental setup with four increments:

- 1st increment:* Only the six evolving robots were present [Fitness limit: 20].
- 2nd increment:* The three preprogrammed robots were introduced but moving at 25% of full speed [Fitness limit: 100].
- 3rd increment:* The speed of the three preprogrammed robots was increased to 50% of full speed [Fitness limit: 200].
- 4th increment:* The speed of the three preprogrammed robots was increased to full speed.

We conducted 30 evolutionary runs till the 2000th generation. The average fitness obtained in a post evaluation (100 samples) of the best chromosome from each run was 358. The average fitness score obtained in 100 samples with a homogeneous group of nine preprogrammed robots was 363. The average post evaluation fitness obtained by the larger mixed groups was thus slightly lower than the fitness obtained by nine preprogrammed robots. However, 12 out of the 30 evolutionary runs produced solutions for mixed groups that obtained a higher post evaluation fitness than nine preprogrammed robots (Mann-Whitney,  $p < 0.02$ ).



The average post evaluation fitness of the best mixed group was 403, thus well above the score obtained by a homogeneous group of nine preprogrammed robots.

We also observed collaborated between the six evolved robots and the three preprogrammed robots just like in our previous experiments. For the best solution evolved, the average distance from the center of the nest to each of the preprogrammed robots was 0.58 m, whereas the average distance to each of the evolved robots was 1.23 m.

## Conclusions

In this paper, we evaluated how the presence of preprogrammed robots affects the evolutionary process and the behaviors evolved in a multirobot foraging task. We conducted evolutions in which a preprogrammed robot was present and evolutions in which it was absent. Without the preprogrammed robot, three different kinds of foraging behaviors were evolved: one in which robots push prey to the nest, one in which robots continually pickup and drop prey, and one (much more efficient) in which robots pickup and carry prey to the nest.

In the setup in which the preprogrammed robot was present, we only observed the pickup and carry behavior. To increase the rate at which foraging solutions are evolved, we conducted a series of incremental evolution experiments in which the preprogrammed robot initially moved at a lower speed and only after the evolved robots had learned to forage did the preprogrammed robot start to move at normal speed. We applied a similar incremental approach for a mixed group of nine robots. We found that when preprogrammed robots were present, the highest performing evolving robots had learned to collaborate with them: the evolving robots targeted prey far from the nest and dropped them close to the nest for the preprogrammed robots to pickup and deploy in the nest. As a result, the robots occupied different regions of the environment and avoided collisions.

The results demonstrate that robots can be evolved to collaborate with preprogrammed robots. The evolving robots did not adopt neither the preprogrammed solution nor the preprogrammed communication protocol, but instead assumed different roles and collaborated with the preprogrammed robots.

In this study, the preprogrammed robots had a complete solution: they were able to forage on their own. In ongoing work, we are evolving robots to fill in the behavioral gaps between robots preprogrammed with different partial solutions to complex tasks.

**Acknowledgements:** This study was supported by the FCT grant PTDC/EEACRO/104658/2008.

## References

Ampatzis, C., Tuci, E., Trianni, V., Christensen, A. L., and Dorigo, M. (2009). Evolving self-assembly in au-

tonomous homogeneous robots: experiments with two physical robots. *Artificial Life*, 15(4):465–484.

Beer, R. D. and Gallagher, J. C. (1992). Evolving dynamical neural networks for adaptive behavior. *Adaptive Behavior*, 1:91–122.

Cliff, D., Husbands, P., and Harvey, I. (1993). Explorations in evolutionary robotics. *Adaptive Behavior*, 2(1):73.

Şahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. In *Swarm Robotics Workshop: State-of-the-art Survey*, pages 10–20. Springer-Verlag, Heidelberg, Germany.

Dorigo, M., Trianni, V., Şahin, E., Groß, R., Labella, T., Baldassarre, G., Nolfi, S., Deneubourg, J., Mondada, F., Floreano, D., and Gambardella, L. M. (2004). Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots*, 17(2):223–245.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.

Nolfi, S. and Floreano, D. (1998). Coevolving predator and prey robots: Do "arms races" arise in artificial evolution? *Artificial Life*, 4(4):311–335.

Nolfi, S. and Floreano, D. (2000). *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. MIT Press, Cambridge, MA.

Nolfi, S., Floreano, D., Miglino, O., and Mondada, F. (1994). How to evolve autonomous robots: Different approaches in evolutionary robotics. In *Proceedings of the 4th International Workshop on Artificial Life*, pages 190–197. MIT Press, Cambridge, MA.

Quinn, M., Smith, L., Mayley, G., and Husbands, P. (2003). Evolving controllers for a homogeneous system of physical robots: Structured cooperation with minimal sensors. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 361(1811):2321.

Schwefel, H.-P. (1995). *Evolution and Optimum Seeking*. Wiley & Sons, New York.

Trianni, V., Nolfi, S., and Dorigo, M. (2006). Cooperative hole avoidance in a swarm-bot. *Robotics and Autonomous Systems*, 54(2):97–103.

Tuci, E., Ampatzis, C., Vicentini, F., and Dorigo, M. (2008). Evolving homogeneous neurocontrollers for a group of heterogeneous robots: Coordinated motion, cooperation, and acoustic communication. *Artificial Life*, 14(2):157–178.



# Evolutionary dynamics of collective action

Jorge M. Pacheco<sup>1,2</sup>, Francisco C. Santos<sup>3,1</sup>, Max O. Souza<sup>4</sup> and Brian Skyrms<sup>5</sup>

<sup>1</sup> ATP-group, CMAF, Complexo Interdisciplinar, P-1649-003 Lisboa Codex, Portugal.

<sup>2</sup> Departamento de Matemática e Aplicações, Universidade do Minho, 4710 - 057 Braga, Portugal.

<sup>3</sup> Centro de Inteligência Artificial & Departamento de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal.

<sup>4</sup> Departamento de Matemática Aplicada, Universidade Federal Fluminense, Niterói, 24020-140, Rio de Janeiro, Brasil

<sup>5</sup> Logic and Philosophy of Science, School of Social Sciences, University of California at Irvine, Irvine, CA 92612, U.S.A.

email: pacheco@cii.fc.ul.pt

## Abstract

In the natural world, performing a given task which is beneficial to an entire group often requires the cooperation of several individuals of that group who often share the workload required to perform the task. The modeling toolkit to address problems related with the dynamics of collective action and other conflict of interests is game theory, often combined with its dynamical counterpart, Evolutionary Game Theory [1]. In this context, the last decades have witnessed the discovery of key insights into the emergence and sustainability of cooperation at different levels of organization. Special attention has been paid to two-person dilemmas such as the Prisoner's Dilemma (PD), the Snowdrift Game (SG) and the Stag-Hunt game (SH), which constitute powerful metaphors to describe conflicting situations often encountered in the natural and social sciences.

Yet, unlike two-person games, current models of collective action have typically overlooked the necessity of some form of coordination among individuals, pervasive in biological and social collective dilemmas [2]. From social organization to the salvation of the planet against environmental hazards [3, 4], examples abound where a minimum number of individuals, which does not necessarily equal the entire group, must simultaneously cooperate before any outcome (or public good) is produced. With this abstract we intend to discuss the predictions of evolutionary game theory for the emergence of collective action, whenever a minimum threshold of individuals must cooperate simultaneously in a group before any viable public good is achieved. These conclusions were previously reported in Refs [2, 3, 5, 6].

We have concentrated on two of the most important collective dilemmas: the N-person snowdrift game (NSG) [5] and N-person prisoner's dilemma (NPD) [2]. In doing so, we uncover a new framework in which the advantage or not of cooperators depends sensitively on group and population size, as well as on the threshold for collective action. Such interplay leads to rich evolutionary scenarios of simultaneous co-existence and bi-stability, impossible to anticipate based on the traditional assumption of infinite populations, providing valuable insights into the variety and complexity of many person social dilemmas, inescapable especially among Humans.

In addition, it is noteworthy that irrespective of the distinctive features of the N-person Prisoner's dilemma (a defector's dominance dilemma) and the N-person Snowdrift game (a

coexistence game) [5], the existence of a coordination threshold is able to produce a unifying framework associated with a generalized stag-hunt game [2]. Moreover, the necessity of coordination is shown to increase the equilibrium fraction of cooperators, even if this enhancement comes together with a strong dependence on the initial level of cooperation, since coexistence between cooperators and defectors only emerges when a minimum number of cooperators is already present in the population. This result is of particular relevance given that the existence of coordination thresholds constitutes a rule, rather than the exception. In addition, we shall also discuss how the chances of collective cooperation are strongly dependent on the perception that individuals have of the collective risk of failure [3]. In this context, we are able to show how global cooperation is better achieved within *i*) small groups, addressing *ii*) highly risky situations characterized by *iii*) stringent condition to meet goals [3]. This result has strong implications on our current understanding of various collective problems, from collective hunting, voluntary adoption of public health measures and other prospective choices, to the mitigation of the effects of global warming. Overall, our results reinforce the idea that even minor differences in the nature of collective rewards and/or costs can have a profound effect in the final outcome of evolution.

## References

1. Sigmund, K.: The Calculus of Selfishness. Princeton University Press (2010)
2. Pacheco, J.M., Santos, F.C., Souza, M.O., Skyrms, B.: Evolutionary dynamics of collective action in N-person stag hunt dilemmas. *Proc. R. Soc. B* **276** (2009) 315-321
3. Santos, F.C., Pacheco, J.M.: Risk of collective failure provides an escape from the tragedy of the commons. *Proc Natl Acad Sci U S A* (2011, in press) published online before print, June 9, 2011, doi: 2010.1073/pnas.1015648108.
4. Milinski, M., Sommerfeld, R.D., Krambeck, H.J., Reed, F.A., Marotzke, J.: The collective-risk social dilemma and the prevention of simulated dangerous climate change. *Proc Natl Acad Sci U S A* **105** (2008) 2291-2294
5. Souza, M.O., Pacheco, J.M., Santos, F.C.: Evolution of Cooperation under N-person Snowdrift Games. *J Theo Bio* **260** (2009) 581-588
6. Pacheco, J.M., Santos, F.C., Souza, M.O., Skyrms, B.: Evolutionary dynamics of collective action. In: Chalub, F.A.A.C., Rodrigues, J.F. (eds.): The Mathematics of Darwin's Legacy. Birkhauser, Basel (2011)

# The Influence of Whole Genome Duplication and Subsequent Diversification on Environmental Robustness and Evolutionary Innovation in Gene Regulatory Networks

Qinxin Pan<sup>1</sup>, Christian Darabos<sup>1</sup>, Anna L. Tyler<sup>1</sup>, Jason H. Moore<sup>1</sup>, and Joshua L. Payne<sup>1</sup>

<sup>1</sup>Dartmouth College, Hanover, NH 03755  
Jason.H.Moore@Dartmouth.edu

## Abstract

In biological systems, whole genome duplication and subsequent diversification constitute powerful mechanisms for the discovery of new phenotypes and for the protection of these phenotypes against environmental perturbation. Here, we use Random Boolean Networks to investigate the influence of these genetic mechanisms on the relationship between evolutionary innovation and environmental robustness in gene regulatory networks. We find that whole genome duplication is highly deleterious in ancestral environments, but provides fitness advantages in novel environments, which come at the cost of reduced environmental robustness. We then show that the subsequent diversification of duplicated networks, via the loss of regulatory interactions, can partly negotiate this trade-off, improving evolutionary innovation and environmental robustness. We conclude by discussing the implications, limitations, and future directions of our research.

## Introduction

Biological systems exhibit two crucial and seemingly antagonistic properties: robustness and evolvability (Wagner, 2005). Regardless of the level of biological organization, living organisms display remarkable resilience to changing conditions, and at the same time, they are able to respond to these changes by developing novel phenotypes. At first glance, these qualities seem paradoxical, yet both empirical (Bloom et al., 2006; Ferrada and Wagner, 2008; Isalan et al., 2008) and theoretical (Aldana et al., 2007; Wagner, 2008; Draghi et al., 2010) analyses suggest their compatibility.

The relationship between robustness and evolvability has been investigated in biological systems ranging in scale from the molecule (Schuster et al., 1994; Cowperthwaite et al., 2008) to the cell (Aldana et al., 2007; Ciliberti et al., 2007a,b). At the cellular level, gene expression patterns are robust to changing environmental conditions, such as alterations in growth medium or the concentration levels of transcription factors (Alon, 2007). This insensitivity to environmental perturbation is largely influenced by the structure of the underlying gene regulatory network (GRN) (Aldana and Cluzel, 2003). A GRN consists of a set of genes, represented as vertices, linked by directed edges if the gene-product (e.g., protein, mRNA, microRNA) of the source gene has

a regulatory influence on the target gene. Recent analyses of model GRNs have revealed that robustness is often correlated with the capacity for evolutionary innovation (Ciliberti et al., 2007a; Aldana et al., 2007).

One major form of structural change in GRNs comes from whole genome duplication (WGD) events, wherein the entire gene repertoire of an organism, including regulatory interactions, is doubled (Sémon and Wolfe, 2007). WGD has long been recognized as a driver of evolutionary innovation (Ohno, 1970) and recent genetic analyses have demonstrated that several major evolutionary transitions resulted from ancient WGD events (Kellis et al., 2004; De Bodt et al., 2005; Taylor et al., 2003). For example, the origin of the budding yeast *Saccharomyces cerevisiae* (Kellis et al., 2004) and the radiation of the angiosperms into over 250,000 species (De Bodt et al., 2005) have both been attributed to WGD. The duplication of genetic material has implications for environmental robustness, as redundant genes diverge to compartmentalize the original function of the ancestral gene (subfunctionalization) (Sémon and Wolfe, 2007). In *S. cerevisiae*, for example, this occurs through the differential expression of redundant genes under various growth conditions (Kafri et al., 2005). WGD also has implications for evolutionary innovation, as duplicate genes diverge to acquire new functions (neofunctionalization) (Sémon and Wolfe, 2007). In *S. cerevisiae*, the ability to consume glucose and grow anaerobically have both been attributed to the genetic diversification that followed a WGD event (Piškur, 2001).

Despite the known importance of WGD events for evolutionary processes, their influence on environmental robustness and evolutionary innovation in GRNs is not thoroughly understood. Here, we use Random Boolean Networks (RBNs) (Kauffman, 1969) to model the dynamics of GRNs. We simulate WGD events in RBNs and quantify their effect on environmental robustness and evolutionary innovation.

This paper is structured as follows. In the subsequent section, we present the key concepts of this work. We then present our model and the details of our simulations, ana-

lyze and discuss our results, and conclude with an outline of future research directions.

## Background

### Random Boolean Networks

Random Boolean Networks (RBNs) are abstract dynamical models of gene regulatory networks (GRNs) (Kauffman, 1969). RBNs consist of  $N$  nodes, which represent genes, and directed edges, which represent regulatory interactions. Node states are binary, representing the expression (1) or repression (0) of gene products. Node states are also dynamic, such that the state of a node in time step  $t + 1$  is dependent upon the states of its regulating nodes in time step  $t$ . To model this dependence, each vertex is associated with a Boolean update function, which is captured by a look-up table that explicitly maps the output expression state for all possible combinations of input states. These output expression states are drawn at random with probability  $p_{\text{expr}}$  and are held fixed throughout the system's dynamics.

Node states are updated synchronously and in discrete time. The dynamics of a RBN begin with a prespecified initial configuration of node states, which represents regulatory factors upstream of the GRN (Ciliberti et al., 2007a). After at most  $2^N$  time steps, the system will encounter a configuration previously visited, thus entering a cycle of one or more configurations, which is referred to as an attractor.

An important aspect of RBNs is that their dynamical behavior falls into one of three regimes: ordered, critical, or chaotic. Systems in the ordered regime exhibit short attractors that are relatively insensitive to environmental perturbation. At the other end of the spectrum, systems in the chaotic regime possess longer attractors that are highly sensitive to environmental perturbation. The critical regime lies at the transition between the ordered and chaotic regimes, offering a balance between the ability to withstand environmental perturbation (robustness) and the ability to utilize these perturbations for evolutionary innovation (evolvability) (Aldana et al., 2007).

### WGD and Subsequent Diversification

Immediately following whole genome duplication (WGD), organismal stability is generally reduced, leading to a decrease in fitness (van Hoek and Hogeweg, 2009). However, duplicate genes supply new genetic material, which can be shaped via mutation and selection to produce novel functions. These functions may allow for more rapid adaptation if a new environment is encountered, providing potential fitness benefits (van Hoek and Hogeweg, 2009). The genetic reorganization that accompanies such diversification may occur via gene loss, gene rearrangements, or alterations in the circuitry of genetic regulation (Sémon and Wolfe, 2007).

## Methods

In this section, we separately present our implementations of RBN generation, duplication, and diversification. We then quantify environmental robustness and evolutionary innovation, outline the evolutionary processes used in our analyses, and provide the details of our simulations.

### RBN Topology

The degree distribution of a RBN has an important influence on system dynamics (Aldana and Cluzel, 2003; Oikonomou and Cluzel, 2006; Aldana et al., 2007). Here, we consider RBNs with Poisson input degree distributions and power-law output degree distributions, as empirical evidence suggests that such topologies are representative of the GRNs of several organisms (Aldana and Cluzel, 2003; Albert, 2005). RBN topologies are generated as described by Darabos et al. (2009).

### Duplication

WGD is simulated by first creating a mirror-image of the original RBN and then linking the duplicate and original components by drawing edges from the source nodes in one component to the targets in the other (Fig. 1a,b). Each node in the duplicated RBN has twice as many inputs as the corresponding node in the non-duplicated RBN. As a result, the number of entries in the look-up table is squared. To populate the entries of each table, we follow Aldana et al. (2007): when the duplicate regulatory inputs are not expressed, the Boolean rules remain identical to those prior to duplication. However, when the duplicate regulatory inputs are expressed the Boolean rules are assigned at random with probability  $p_{\text{expr}}$ .

### Diversification

To simulate the genetic diversification that follows a WGD event, we take a conservative approach and assume that only regulatory interactions can be lost (akin to structural simplification algorithms for neural networks (Le Cun et al., 1990)). This represents a mutation to the promoter region of a gene that prohibits the binding of one of its regulating gene products. While this type of mutation represents only a small subset of all possible forms of genetic reorganization, it offers a useful and parsimonious starting point. Further, empirical data suggest that (i) interactions are lost at a rate that is three orders of magnitude larger than the rate at which they are gained (Wagner, 2001) and (ii) rates of alternative forms of reorganization, such as gene loss, are significantly reduced among transcription factors (De Bodt et al., 2005), which are the primary gene products modeled by RBNs.

In our simulations, diversification occurs through the removal of all non-functional regulatory edges (Fig. 1c,d). These edges link a source to a target, where the state of the source does not influence the expression of the target. Such edges are referred to as canalized (Kauffman et al., 2004),

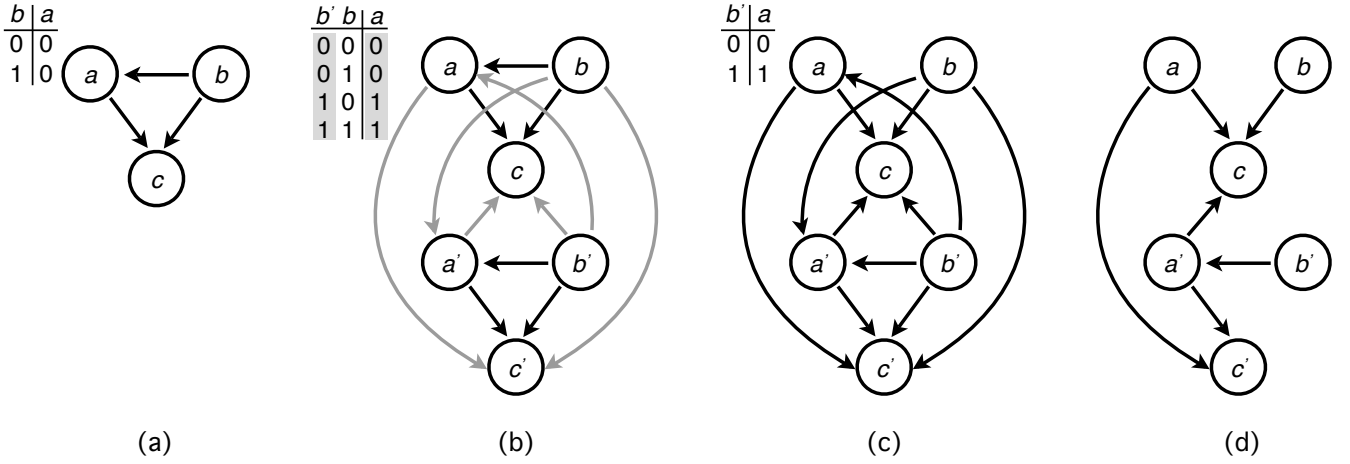


Figure 1: Schematic of whole genome duplication (WGD) and the subsequent diversification of a Random Boolean Network (RBN). The (a) non-duplicated RBN (b) undergoes WGD, wherein its entire gene repertoire is copied. Additional edges (gray) are drawn from the source nodes of one component to the target nodes of the other. The look-up tables of each node are expanded as described by Aldana et al. (2007). As an illustrative example, we depict one possible expansion of the look-up table of node  $a$ . (c) Diversification occurs via edge loss. E.g.,  $a \leftarrow b$ . (d) All edges are removed where the state of the target node is independent of the state of the source node. The diversification process continues throughout the evolution of the population, resulting in RBN topologies that differ markedly from those that immediately followed WGD.

and their removal does not immediately affect the dynamics of the RBN.

## Environmental Robustness

Environmental perturbations come in many forms, including alterations in temperature, growth medium, or biotic environment. A RBN is environmentally robust if its phenotype is insensitive to these non-genetic perturbations. We measure environmental robustness as the sensitivity of a RBN to the perturbation of a single, randomly chosen configuration of its attractor. Specifically, we systematically perturb the state of each node in the randomly chosen configuration, one at a time, and measure the proportion of perturbations in which the RBN returns to its original attractor.

## Evolutionary Innovation

An evolutionary innovation can be thought of as a change in phenotype that confers a fitness advantage. To assess evolutionary innovation, we measure the fitness of a RBN as the ability of its attractor to match a randomly generated target attractor. This target attractor represents the gene expression pattern required for optimal adaptation to a given environment. Fitness thus provides a proxy for evolutionary innovation.

For each RBN, we randomly select a single output node and record the sequence of output states  $\sigma_{\text{out}}$  during its attractor. The fitness  $F$  of a RBN is then calculated as the Hamming distance between the output and target sequences

(Oikonomou and Cluzel, 2006),

$$F = \max \left\{ 1 - \frac{1}{\text{lcm}(L, L_c)} \sum_{t=1}^{\text{lcm}(L, L_c)} |\sigma_{\text{out}}(t) - \sigma_{\text{target}}(t)| \right\}, \quad (1)$$

where  $L$  is the length of the output sequence,  $L_c$  is the length of the target sequence, and  $\text{lcm}$  denotes the least common multiple. To facilitate the comparison of sequences with  $L \neq L_c$ , both sequences are concatenated onto themselves until they are of length  $\text{lcm}(L, L_c)$ . To ensure that fitness is independent of the starting position of the output sequences, we take the maximum fitness over all cyclic permutations of  $\sigma_{\text{out}}$ .

## Evolution

We simulate the evolution of randomly initialized populations of RBNs in discrete, non-overlapping generations. In every generation, the fitness of each RBN is assessed according to Eq. 1. RBNs are then selected with uniform probability, with replacement, to compete in binary tournaments. Within a tournament, the RBN with the highest fitness is selected to move on to the next generation, after undergoing mutation. Mutation only affects the RBN's look-up tables, such that the entries in the look-up tables associated with each vertex undergo bit-flip mutation with probability  $p_{\text{mut}}$ . This process of selection and mutation is repeated until the next generation is fully populated.

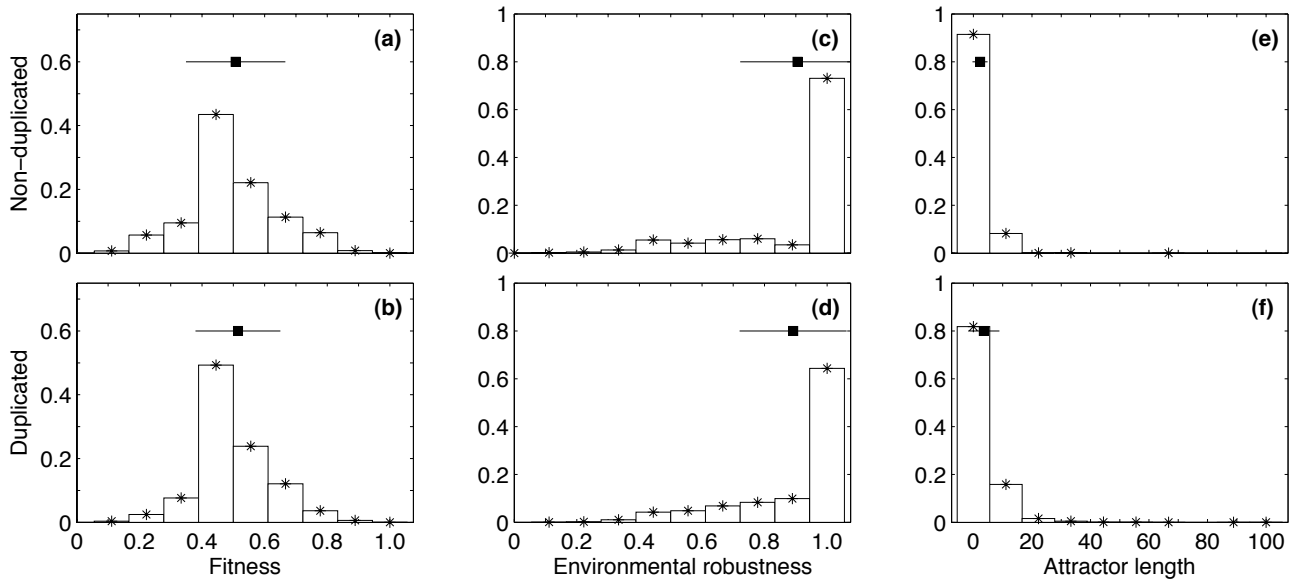


Figure 2: (a,b) Fitness, (c,d) environmental robustness, and (e,f) attractor length of non-duplicated (top row) and duplicated (bottom row) RBNs in novel environments. Each panel depicts the frequency distribution of data across 10,000 independent replications. The filled squares depict the mean of those data and the horizontal lines depict one standard deviation. The asterisk symbols are placed atop each non-empty bin as a visual aid. In (f), there was a single outlier with an attractor length  $L = 254$ , which is not shown.

## Simulation Details

We consider RBNs with  $N = 10$  nodes prior to duplication and  $N = 20$  nodes after duplication. RBNs are initialized near the critical regime by setting the probability of gene expression to  $p_{\text{expr}} = 0.5$  and the scaling exponent of the output degree distribution to  $\gamma = 1.894$ , which yields criticality in RBNs with  $N = 10$  (Aldana et al., 2007).

Evolutionary analyses are conducted with a population size of 500, wherein each RBN is paired with its own, randomly chosen initial state which does not change throughout the evolutionary trajectory of its lineage. In each experiment, we consider 100 independent replications that each consist of 5,000 generations. Mutation occurs with probability  $p_{\text{mut}} = 0.002$  per look-up table entry. In the experiments that include diversification, the deterministic edge-loss process only occurs every 10 generations, due to computational constraints.

## Results

We present our results in four successive phases. First, we compare the immediate effects of WGD on the fitness of RBNs in their ancestral environments. Second, we compare the immediate effects of WGD on the fitness and robustness of RBNs in novel environments. Third, we consider the evolutionary dynamics of fitness and robustness for non-duplicated and duplicated RBNs. Fourth, we compare the evolutionary dynamics of these same quantities when dupli-

cated RBNs are allowed to undergo diversification.

### WGD in an Ancestral Environment

To simulate an ancestral environment, we simply assume that the expression profile of a randomly generated RBN is optimally adapted. To do this, we choose a random node from the RBN and define it as  $\sigma_{\text{target}}$ . We then simulate a WGD event, designate the expression profile of the same node in the duplicated RBN as  $\sigma_{\text{out}}$ , and compute its fitness (Eq. 1). To collect meaningful statistics, we repeat this process 10,000 times.

WGD is highly deleterious in an ancestral environment. Optimal fitness is maintained in only  $\sim 42\%$  of WGD events. Of the remaining  $\sim 58\%$  of duplicated RBNs, average fitness decreases to  $0.37 \pm 0.002$ .

### WGD in a Novel Environment

To simulate a novel environment, we randomly generate  $\sigma_{\text{target}}$  of length  $L_c = 10$  (Oikonomou and Cluzel, 2006). We then generate a RBN, choose a random node, designate its expression profile as  $\sigma_{\text{out}}$ , and compute the RBN's fitness (Eq. 1). In addition, we measure the RBN's environmental robustness. We then collect these data for the same RBN after WGD. As in the previous analysis, this process is repeated 10,000 times.

Duplicated RBNs exhibit a marginal fitness advantage over their non-duplicated counterparts (Fig. 2a,b; Stu-

dent's t-test,  $p = 5.43 \times 10^{-4}$ ). However, this advantage comes at the expense of a marginal decrease in environmental robustness (Fig. 2c,d; Kolmogorov-Smirnov test,  $p = 3.46 \times 10^{-74}$ ). These subtle differences can be attributed to the increased attractor length of the duplicated RBNs (Fig. 2e,f; Student's t-test,  $p = 8.9 \times 10^{-9}$ ), which have a higher probability of matching  $\sigma_{\text{target}}$  and exhibit a greater sensitivity to perturbation.

### Evolutionary Dynamics of Duplicated RBNs

We now turn from a static analysis of environmental robustness and fitness to an evolutionary analysis of these quantities for populations of non-duplicated and duplicated RBNs. As in the previous section, we consider novel environments by randomly generating target sequences  $\sigma_{\text{target}}$  of length  $L_c = 10$ .

As observed in our previous analysis, the duplicated RBNs have an immediate, albeit slight, fitness advantage in a novel environment (Fig. 3a), but are marginally less robust (Fig. 3b). These differences in fitness and robustness become more pronounced throughout the evolutionary process. Duplicated RBNs reach a plateau of average fitness at  $0.92 \pm 0.006$  (Fig. 3a, squares) while the non-duplicated RBNs stagnate at an average fitness of  $0.89 \pm 0.008$  (Fig. 3a, triangles). Simultaneously, the duplicated RBNs drop to an average environmental robustness of  $0.73 \pm 0.042$  (Fig. 3b, squares), while the non-duplicated RBNs retain a higher environmental robustness of  $0.83 \pm 0.018$  (Fig. 3b, triangles). Thus, WGD in the absence of subsequent diversification leads to a trade-off between environmental robustness and evolutionary innovation.

### Evolutionary Dynamics of Diversified RBNs

To investigate the effects of diversification after WGD, we conduct an evolutionary analysis of paired populations of duplicated RBNs, wherein diversification can only occur in one of the initially identical populations.

Diversification promotes evolutionary innovation, with populations reaching an average fitness of  $0.95 \pm 0.013$  (Fig. 4a, open circles), a significant improvement over the fitness obtained with WGD alone (Fig. 4a, closed squares). Simultaneously, diversification increases environmental robustness (Fig. 4b, open circles), though not to the same levels observed prior to WGD (Fig. 3b, triangles). Thus, diversification allows for the partial negotiation of the trade-off between environmental robustness and evolutionary innovation that is induced by WGD.

The diversification process also leads to appreciable structural changes in RBN topologies, with average connectivity dropping rapidly (Fig. 4a, inset). RBN dynamics are also affected, with attractor lengths of diversified networks settling to an average of  $9.48 \pm 0.174$ , as compared to  $9.92 \pm 0.112$  for non-diversified networks (Fig. 4b, inset). The probability of gene expression  $p_{\text{expr}}$  remains approximately constant

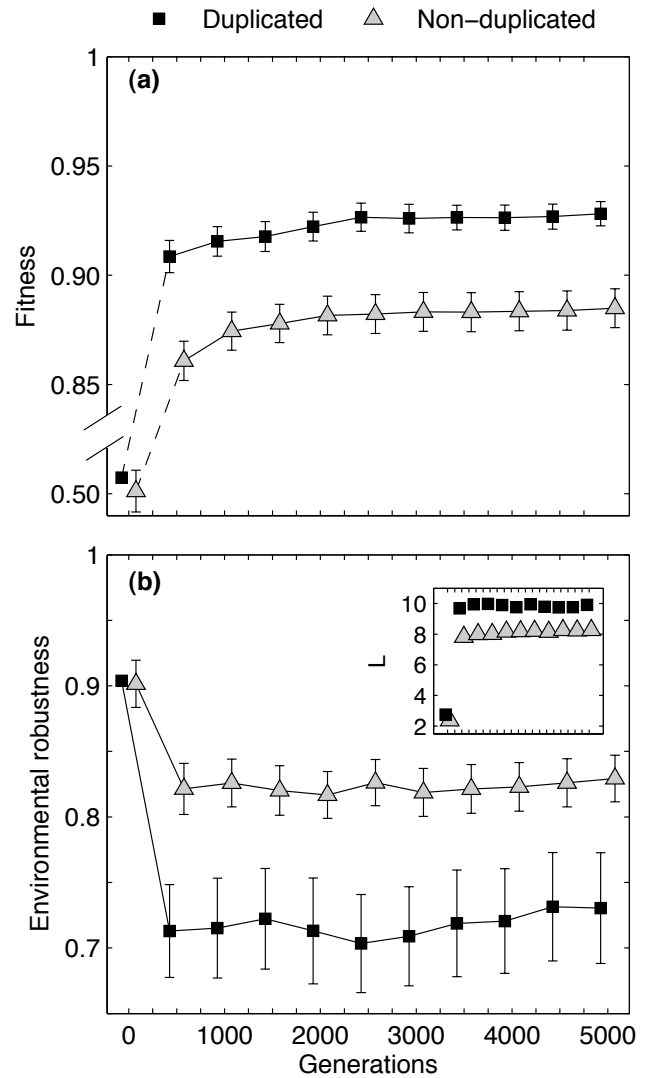


Figure 3: Evolutionary dynamics of (a) fitness and (b) environmental robustness for populations of duplicated and non-duplicated RBNs in novel environments. Data represent the mean of 100 independent replications and error-bars denote a single standard deviation. The inset in (b) depicts the average attractor length  $L$ . Data are deliberately offset in the horizontal dimension for visual clarity. Note the break in scale on the y-axis of (a). The scale of the x-axis is the same in all panels, including insets.

at 0.5 throughout the evolutionary process (data not shown).

### Discussion

We have used Random Boolean Networks (RBNs) to investigate the influence of whole genome duplication (WGD) and subsequent diversification on evolutionary innovation and environmental robustness in gene regulatory networks (GRNs). There are some limitations to our approach that

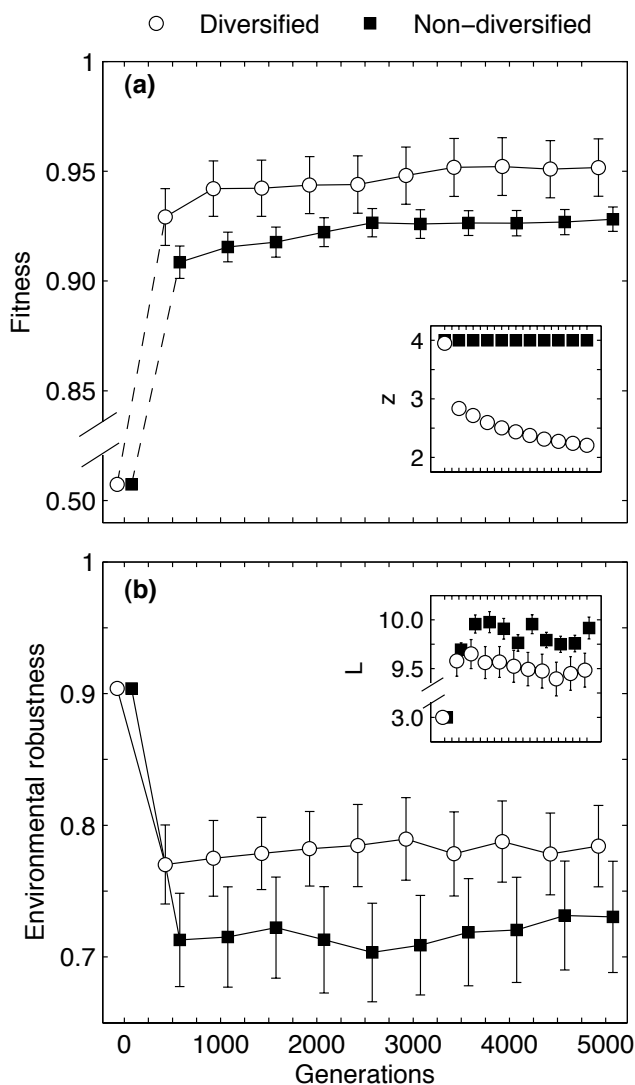


Figure 4: Evolutionary dynamics of (a) fitness and (b) environmental robustness for populations of RBNs following WGD in novel environments, with (open circles) and without (closed squares) subsequent diversification. Data represent the mean of 100 independent replications and error-bars denote a single standard deviation. The inset in (a) depicts the average network connectivity  $z$  and the inset in (b) depicts the average attractor length  $L$ . Data are deliberately offset in the horizontal dimension for visual clarity. Note that the data represented by the closed squares are the same as in Fig. 3. Also note the break in scale on the y-axis of (a) and the inset of (b). The scale of the x-axis is the same in all panels, including insets.

are worth highlighting. First, while the genotype produced by a WGD event is purely redundant, its resulting phenotype may differ immediately from that of the non-duplicated genotype. This occurs because both the original and du-

plicated nodes acquire new regulatory connections, necessitating the random initialization of entire segments of the expanded look-up tables. Further, the duplicated node cannot, under most circumstances, act as a “backup” because alterations to the original node may lead to phenotypic alterations that are not easily compensated for by the duplicate. Second, because many phenotypes yield identical fitness, and the phenotypic contribution of a single gene cannot be separated from its interaction partners, it is not possible to discern whether the observed changes in environmental robustness and evolutionary innovation are due to subfunctionalization, neofunctionalization, or a combination thereof (He and Zhang, 2005).

Despite these limitations, our analyses have helped to clarify the influence of WGD and subsequent diversification on environmental robustness and evolutionary innovation in GRNs. While deleterious in ancestral environments, WGD provided marginal fitness benefits in novel environments, coming at the expense of reduced environmental robustness (Fig. 2). Over evolutionary time, these differences magnified, with duplicated RBNs achieving significantly higher fitness and significantly lower environmental robustness than their non-duplicated counterparts (Fig. 3). Genetic diversification, via the loss of non-functioning regulatory interactions, was able to partly negotiate this trade-off, leading to improvements in both fitness and environmental robustness (Fig. 4).

Environmental robustness and evolutionary innovation were therefore inversely related in this system. This occurred because fitness assignment was based solely on the ability of a RBN to match a target expression profile  $\sigma_{\text{target}}$  (Eq. 1). This induced selection pressure for longer attractors (insets in Figs. 3b and 4b), because increasing the duration of the expression profile of the output node increased the probability that some segment of that profile matched  $\sigma_{\text{target}}$ . In turn, environmental robustness decreased, because longer attractors were more sensitive to perturbation. Thus, while some aspects of robustness and evolvability are positively correlated in RBNs (Aldana et al., 2007), robustness to environmental perturbation and the ability to match a target phenotype are not amongst them.

Diversification increased environmental robustness (Fig. 4b) through a reduction in network connectivity (Fig. 4a, inset). This shifted the RBN dynamics closer to the critical regime and therefore reduced the average attractor length (Fig. 4b, inset), yielding more environmentally robust attractors. It is notable that this reduction in attractor length did not lead to a corresponding reduction in fitness (Fig. 4a). How the diversified RBNs were able to attain increased fitness using shorter attractors is not yet known. An analysis of the structural properties of evolved RBNs, such as network excitation (Draghi and Wagner, 2009) or degree distribution (Aldana et al., 2007), may provide more insight into the mechanisms by which diversification can simultaneously



increases fitness and environmental robustness.

In the absence of diversification, the selective advantage of WGD may depend heavily on the frequency with which environmental perturbations occur. Selection may favor phenotypes that consistently yield expression profiles of average fitness over those that inconsistently yield expression profiles of high fitness. By placing non-duplicated and duplicated RBNs in a head-to-head competition under varying levels of environmental perturbation, future work will seek to determine how selection moderates the trade-off between environmental robustness and evolutionary innovation, and to discover the conditions under which selection leads to the “survival of the flattest” (Wilke et al., 2001).

The environments considered in this study were static, meaning that the target gene expression profile did not change over time. Several studies have demonstrated the importance of dynamic environments in shaping a population’s potential for evolutionary innovation (Kashtan et al., 2007; Draghi and Wagner, 2009). Future work will seek to understand how WGD and subsequent diversification influence evolutionary innovation and robustness in dynamic environments.

Future work will also seek to expand upon our usage of fitness as a proxy for evolutionary innovation. It may prove insightful to analyze not only the ability to move toward a specific fitness optimum, but also the ability to move toward arbitrary fitness optima. Such measurements of the diversity of accessible phenotypes are common in studies of evolvability (Ciliberti et al., 2007a; Cowperthwaite et al., 2008; Wagner, 2008), and could be incorporated into our analysis (Draghi and Wagner, 2009). In addition, we will also investigate alternative forms of genetic diversification, with a particular focus on gene loss, which will allow for a more direct comparison with alternative models of WGD and diversification in GRNs (Wagner, 1996). These extensions, among others, will lead to a more thorough understanding of how various genetic mechanisms influence the relationship between robustness and evolvability in gene regulatory networks.

## Acknowledgements

This work was partially supported by NIH grants R01-LM009012, R01-LM010098, and R01-AI59694. C.D. was supported by the Swiss National Science Foundation, grant PBLAP2-129416/1. J.L.P. was supported by NIH grant K25-CA134286.

## References

- Albert, R. (2005). Scale-free networks in cell biology. *Journal of Cell Science*, 118:4947–4957.
- Aldana, M., Balleza, E., Kauffman, S., and Resendiz, O. (2007). Robustness and evolvability in genetic regulatory networks. *Journal of Theoretical Biology*, 245:433–448.
- Aldana, M. and Cluzel, P. (2003). A natural class of robust networks. *Proceedings of the National Academy of Sciences*, 100:8710–8714.
- Alon, U. (2007). *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman and Hall.
- Bloom, J. D., Labthavikul, S. T., Otey, C. R., and Arnold, F. H. (2006). Protein stability promotes evolvability. *Proceedings of the National Academy of Sciences*, 103:5869–5874.
- Ciliberti, S., Martin, O. C., and Wagner, A. (2007a). Innovation and robustness in complex regulatory gene networks. *Proceedings of the National Academy of Sciences*, 104:13591–13596.
- Ciliberti, S., Martin, O. C., and Wagner, A. (2007b). Robustness can evolve gradually in complex regulatory gene networks with varying topology. *PLoS Computational Biology*, 3:e15.
- Cowperthwaite, M. C., Economo, E. P., Harcombe, W. R., Miller, E. L., and Meyers, L. A. (2008). The ascent of the abundant: how mutational networks constrain evolution. *PLoS Computational Biology*, 4:e10000110.
- Darabos, C., Tomassini, M., and Giacobini, M. (2009). Dynamics of unperturbed and noisy generalized Boolean networks. *Journal of Theoretical Biology*, 260:531–544.
- De Bodt, S., Maere, S., and Van de Peer, Y. (2005). Genome duplication and the origin of the angiosperms. *TRENDS in Ecology and Evolution*, 20:591–597.
- Draghi, J. and Wagner, G. (2009). The evolutionary dynamics of evolvability in a gene network model. *Journal of Evolutionary Biology*, 22:599–611.
- Draghi, J. A., Parsons, T. L., Wagner, G. P., and Plotkin, J. B. (2010). Mutational robustness can facilitate adaptation. *Nature*, 463:353–355.
- Ferrada, E. and Wagner, A. (2008). Protein robustness promotes evolutionary innovations on large evolutionary time-scales. *Proceedings of the Royal Society London B*, 275:1595–1602.
- He, X. and Zhang, J. (2005). Rapid subfunctionalization accompanied by prolonged and substantial neofunctionalization in duplicate gene evolution. *Genetics*, 169:1157–1164.
- Isalan, M., Lemerle, C., Michalodimitrakis, K., Horn, C., Beltrao, P., Raineri, E., Garriga-Canut, M., and Serrano, L. (2008). Evolvability and hierarchy in rewired bacterial gene networks. *Nature*, 452:840–846.
- Kafri, R., Bar-Even, A., and Pilpel, Y. (2005). Transcription control reprogramming in genetic backup circuits. *Nature Genetics*, 37:295–299.
- Kashtan, N., Noor, E., and Alon, U. (2007). Varying environments can speed up evolution. *Proceedings of the National Academy of Sciences*, 104:13711–13716.
- Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22:437–467.

- Kauffman, S. A., Peterson, C., Samuelsson, B., and Troein, C. (2004). Genetic networks with canalizing Boolean rules are always stable. *Proceedings of the National Academy of Sciences*, 101:17102–17107.
- Kellis, M., Birren, B. W., and Lander, E. S. (2004). Proof and evolutionary analysis of ancient genome duplication in the yeast *Saccharomyces cerevisiae*. *Nature*, 428:617–624.
- Le Cun, Y., Denker, J. S., and Solla, S. A. (1990). Optimal brain damage. *Advances in Neural Information Processing Systems*, 2:598–605.
- Ohno, S. (1970). *Evolution by Gene Duplication*. George Allen and Unwin, London, UK.
- Oikonomou, P. and Cluzel, P. (2006). Effects of topology on network evolution. *Nature Physics*, 2:532–536.
- Piškur, J. (2001). Origin of the duplicated regions in the yeast genomes. *Trends in Genetics*, 17:302–303.
- Schuster, P., Fontana, W., Stadler, P. F., and Hofacker, I. L. (1994). From sequences to shapes and back: a case study in RNA secondary structures. *Proceedings of the Royal Society London B*, 255:279–284.
- Sémon, M. and Wolfe, K. H. (2007). Consequences of genome duplication. *Current Opinion in Genetics and Development*, 17:505–512.
- Taylor, J. S., Braasch, I., Frickey, T., Meyer, A., and Van de Peer, Y. (2003). Genome duplication, a trait shared by 22,000 species of ray-finned fish. *Genome Research*, 13:382–390.
- van Hoek, M. J. A. and Hogeweg, P. (2009). Metabolic adaptation after whole genome duplication. *Molecular Biology and Evolution*, 26:2441–2453.
- Wagner, A. (1996). Genetic redundancy caused by gene duplications and its evolution in networks of transcriptional regulators. *Biological Cybernetics*, 74:557–567.
- Wagner, A. (2001). The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes. *Molecular Biology and Evolution*, 18:1283–1292.
- Wagner, A. (2005). *Robustness and Evolvability in Living Systems*. Princeton University Press.
- Wagner, A. (2008). Robustness and evolvability: a paradox resolved. *Proceedings of the Royal Society London B*, 275:91–100.
- Wilke, C. O., Wang, J. L., Ofria, C., Lenski, R. E., and Adami, C. (2001). Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, 412:331–333.

# Homologous and Nonhomologous Rearrangements: Interactions and Effects on Evolvability

David P. Parsons<sup>1,3</sup>, Carole Knibbe<sup>2,3</sup> and Guillaume Beslon<sup>1,3</sup>

<sup>1</sup>Université de Lyon, CNRS, INRIA, INSA-Lyon, LIRIS, UMR5205, F-69621, France

<sup>2</sup>Université de Lyon, CNRS, INRIA, Université Lyon 1, LIRIS, UMR5205, F-69622, France

<sup>3</sup>IXXI, Institut Rhône-Alpin des Systèmes Complexes, Lyon, F-69007, France

guillaume.beslon@liris.cnrs.fr

## Abstract

By using Aevol, a simulation framework designed to study the evolution of genome structure, we investigate the effect of homologous rearrangements on the course of evolution. We designed an efficient model of rearrangements based on an intermittent search algorithm. Then, using experimental *in silico* evolution, we explore the effect of rearrangement rates on the genome structure. We show that the effect of homologous rearrangements is quite complex. At first glance they appear to be dangerous enough to trigger an indirect selective pressure leading to short genomes when the rearrangement rate is high. However, by analyzing the successful lineage in the best runs, we found that there is a positive correlation between the number of homologous rearrangements and the fitness improvement in these lineages. Thus the impact of homologous rearrangements on evolution is rather complex: dangerous on the one hand but necessary on the other hand, to ensure a sufficient level of evolvability to the organisms. Moreover, our results show that the spontaneous rate of small mutations influences the relative proportions of homologous versus nonhomologous rearrangements.

## Introduction

Chromosomal rearrangements are known to play a major role in evolution. Their most visible effects are quite straightforward: duplications and deletions account for numerous gene acquisitions or losses while translocations and inversions have a direct influence on gene order. However, these direct effects are flanked by other indirect selective pressures. The rates and mechanisms of rearrangements indeed influence the evolvability (Kirschner and Gerhart, 1998) of the lineage and, as it was stated by Earl and Deem (2004), evolvability itself can be subject to evolution. In the long term, more evolvable lineages are more likely to produce beneficial mutations and hence to overcome lineages with lower evolvability. Similarly, Wilke et al. (2001) showed a second-order selective pressure on mutational robustness. The selection of a specific level of evolvability or robustness is said to be indirect because they do not influence the fitness of the organism, but that of its descendants.

Unraveling these second-order pressures is a very challenging matter. Indeed, the underlying processes are complex and act on a very long time scale. It is hence difficult

to tackle such questions either *in vivo* or *in vitro*. Comparative genomics approaches are a way to circumvent this difficulty. However, they are based upon the static snapshots of the contemporary sequences and have to *infer* their evolutionary past.

Artificial life and *in silico* simulations are very useful in such cases, providing us with insights into complex mechanisms and shedding light onto second-order pressures that would have been difficult to identify otherwise (Wilke et al., 2001; Adami, 2006; Misevic et al., 2006; Knibbe et al., 2007; Beslon et al., 2010). They offer a dynamic view of the evolutionary process and provide the experimentalist with a very good control over parameters as well as a perfect fossil record throughout the evolution.

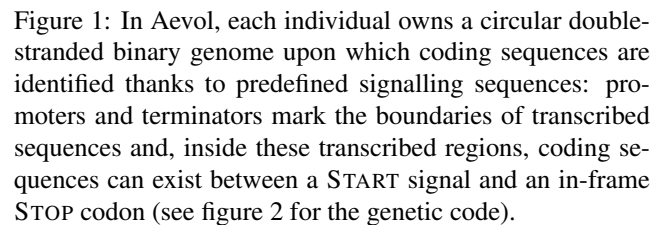
The Aevol model was developed specifically to study the evolution of genome structure. Experiments using this model underlined the major importance of chromosomal rearrangements in the evolutionary process. For a start, we observed that in total absence of chromosomal rearrangements, evolution can hardly occur at all because gene duplications are necessary to acquire new genes and thus new functions. Secondly, it has been shown that, because of rearrangements, non-coding sequences can become mutagenic for the surrounding genes. The consequence is a clear trend for organisms having evolved under high rearrangement rates to own shorter and denser genomes than those having evolved under lower rates of rearrangement (Knibbe et al., 2007; Parsons et al., 2010). As we have already shown, this effect is the consequence of the long-term selection of a specific level of mutational variability (Knibbe et al., 2007).

Unlike point mutations and indels that produce local variations, chromosomal rearrangements can involve huge sequences and turn a very fit individual into an ill-adapted one in a single event. Chromosomal rearrangements can hence be very dangerous. However, rearrangements are usually not fully random. Most rearrangements are the consequence of error-repair mechanisms such as the RecA mediated double strand break repair mechanism (Neidhardt, 1996). These mechanisms usually require that the sequences be similar (at least around the breakpoints) to be rearranged. Such rear-

After an overall presentation of the AevoL model, focusing particularly on the way we take sequence homologies into account in the rearrangement process, we will present our results regarding the different effects of homologous and nonhomologous rearrangements. We will discuss the intricate relationship that exists between homologous and non-homologous rearrangements, and their impact on evolvability.

The AevoI model was developed in our team to study the evolution of genome structure. It simulates the evolution of a population of  $N$  artificial haploid organisms with flexible genomes. Although a description of the model has already been published (see Knibbe et al. (2008) and its supp. mat.), we thereafter provide an overview of the most important principles that are necessary to have a good understanding of the results presented here.

At each generation,  $N$  new individuals are created by reproducing preferentially the best individuals of the parental generation which is then completely replaced. During the replication process, the chromosome can undergo different kinds of modifications: local mutations (point mutations, small insertions and small deletions), but also large chromosomal rearrangements (duplications, deletions, translocations and inversions). At the beginning of the run, all the organisms are initialized with the same random sequence (of 5,000 base-pairs here) which contains at least one gene.



**Transcription** In prokaryotes, transcription initiates at particular sites, called promoters, where the RNA-polymerases recognize a consensus sequence to which they can bind and begin the RNA synthesis. In Aevol, we defined a long consensus sequence, a promoter being a sequence whose Hamming distance  $d$  with this consensus is less than or equal to  $d_{max}$ . In the experiments presented here, the consensus was a 22-base-pairs (bp) sequence and up to  $d_{max} = 4$  mismatches were allowed. This consensus sequence is long enough to ensure that random, non-coding sequences have a low probability to become coding by a single mutation event.

The expression level  $e$  of an RNA is determined according to its promoter sequence:  $e = 1 - \frac{d}{d_{max} + 1}$ . This modulation of the expression level models in a simplified way the basal interaction of the RNA polymerase with the promoter, without additional regulation. It provides duplicated genes with a way to reduce temporarily their phenotypic contribution while diverging toward other functions.

**Translation** Transcribed sequences (RNAs) do not necessarily result in a protein. The translation process of an RNA takes place when a Shine-Dalgarno-like sequence is found, followed, a few base-pairs away, by a START codon (see genetic code on figure 2). Whenever this signal 011011\*\*\*\*000 is found, the following sequence is read three bases (one codon) at a time until the STOP codon (001) is found on the same reading frame. Each codon lying be-

```

foreach Generation do
  // Evaluation
  foreach Individual do
    Identify coding sequences
    foreach CodingSequence do
      | Translate into abstract protein
    end
    Compute phenotype by combining protein
    contributions
    Compute fitness by comparing the phenotype to
    the environmental target
  end

  // Selection
  Sort the individuals by fitness
  Compute the probabilities of reproduction
  Draw the actual numbers of offspring

  // Reproduction
  foreach Individual do
    foreach Offspring do
      | Do Rearrangements
      | Do Local Mutations
    end
  end

  Replace current population
end

```

**Algorithm 1:** Aevol General Algorithm

tween the initiation and termination signals is translated into an abstract “Amino-Acid” using an artificial genetic code, therefore giving rise to the protein’s primary sequence (figure 2). As in real organisms, genes can be found on six different reading frames (three on each strand), giving the possibility for the organisms to evolve overlapping genes, which are commonly found in virus and bacteria.

**Protein “folding” and phenotype computation** To model the activity of proteins and the resulting phenotype, we defined a simple “artificial chemistry” (Dittrich et al., 2001) that describes the organism’s metabolism in a mathematical language. In our simplified artificial world, we assume that there is an abstract, one-dimensional space  $\Omega = [0, 1]$  representing all the possible metabolic processes (that is, in this model, a metabolic process is just a real number). In this “metabolic space”, each protein is involved in a subset of processes (either realizing it or preventing other proteins from realizing it) which is described using the fuzzy set formalism: a given protein can be involved in a metabolic process with a possibility degree lying between 0 and 1. A protein is thus fully characterized by a mathematical function that associates a possibility degree to each metabolic process, describing the fuzzy subset of metabolic processes it is involved in. For simplicity, we use piecewise-linear functions with a symmetric, triangular shape (figure 2). In

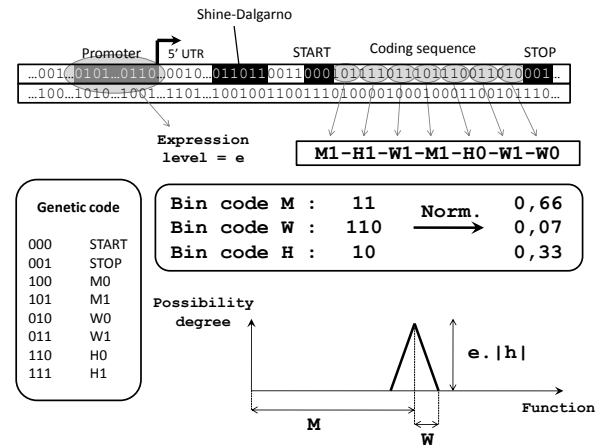


Figure 2: Overview of the transcription-translation-folding process in Aevol. Transcribed sequences are those that start with a promoter (consensus sequence) and end with a terminator sequence (stem-loop structure), not shown on the figure. Coding sequences (genes) are searched within the transcribed sequences; They begin with a Shine-Dalgarno-START sequence and end with a STOP codon. An artificial genetic code (right) is used to convert a gene into the primary sequence of the corresponding protein and a “folding process” enables us to compute the metabolic activity of this protein (functional abilities).

this way, only three numbers are needed to characterize the metabolic activity of a protein: the position  $m$  ( $m \in \Omega$ ) of the triangle on the axis, its half-width  $w$  and its height  $h$  (positive when realizing a function, negative when inhibiting it). This means that the protein contributes to the range  $[m-w, m+w]$  of metabolic processes, with a preference for the processes closest to  $m$  (for which the highest efficiency,  $h$ , is reached). Thus, various types of proteins can co-exist, from highly efficient and specialized ones (small  $w$ , high  $h$ ) to polyvalent but poorly efficient ones (large  $w$ , low  $h$ ).

In this framework, each protein’s primary sequence is decomposed into three interlaced binary subsequences that will in turn be interpreted as the values for the  $m$ ,  $w$  and  $h$  parameters. For instance, the codon 010 (resp. 011) is translated into the single amino acid  $W0$  (resp.  $W1$ ), which means that it contributes to the value of  $w$  by adding a bit 0 (resp. 1) to its binary code. Small mutations in the coding sequence (point mutations, indels, possibly causing frame shifts) will change these parameters, resulting in a modification of the protein’s metabolic activity.

Once all the proteins encoded on the genotype of the organism have been identified and characterized, their activities are combined into a fuzzy set representing the individual’s phenotype  $P = (\cup A_i) \cap (\cup I_j)$ , using Lucasiewicz’ fuzzy operators, with  $A_i$  being the fuzzy subset of the  $i$ -th activating protein ( $h_i > 0$ ) and  $I_j$  the fuzzy subset of the  $j$ -th inhibiting protein ( $h_j < 0$ ). Intuitively, this means that

metabolic processes achieved by the organism are those that are activated and not inhibited. The phenotypic fuzzy set  $P$  indicates to what extent the individual can realize each metabolic process in our abstract metabolic space.

### Environment, adaptation and selection

In Aevol, the environment is represented by a phenotypic target: the fuzzy set  $E$  defined on  $\Omega$  that represents the optimal degree of possibility for each “biological function”. To evaluate an individual, we compare its phenotype  $P$  to the optimal phenotype  $E$ . The “metabolic error”  $g$  is computed as the geometric area between these two sets (figure 3). The lower the metabolic error, the better the individual. This measure penalizes both the under-realization and the over-realization of each function.

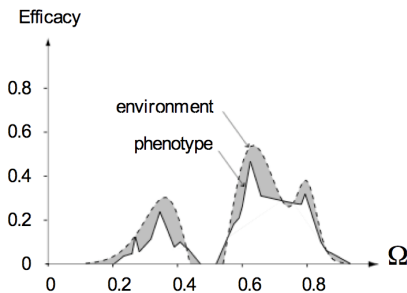


Figure 3: Measure of individual adaptation. Dashed curve: environmental target  $E$ . Solid curve: phenotypic distribution  $P$  (resulting metabolic profile obtained after combining all the proteins). Dark grey filled area: metabolic error  $g$ .

In the current version of Aevol, the population size is constant (here  $N = 1,000$  individuals) and the population is entirely renewed at each generation. A probability of reproduction is assigned to each individual according to its metabolic error and a multinomial drawing determines the actual number of offsprings each individual will have. In the experiments presented here, we used an exponential ranking selection (Blickle and Thiele, 1996). The individuals are sorted by decreasing metabolic error so that the worst individual has rank  $r = 1$  and the best  $r = N$ . The probability of reproduction of an individual is then given by  $\frac{s-1}{s^N-1} s^{N-r}$ , with  $s = 0,998$  being the intensity of selection in all the experiments presented here.

### Genetic operators

During their replication, genomes can undergo different modifications: local mutations (point mutations, insertions or deletions of 1 to 6 bp) and chromosomal rearrangements (duplications, deletions, translocations, inversions).

Mutations and rearrangements affect the genome but do not necessarily have a phenotypic effect. For instance, a mutation that takes place in an untranscribed region will be completely neutral unless it creates a new promoter, which is reasonably rare given the size of the consensus sequence.

The rates at which each type of local mutation occurs are parameters of the model. They are defined as the per-base, per-replication probability of each type of mutation to take place. The chromosomal rearrangement rates however, can not be a direct parameter of the model. Indeed, in this version of the model, a rearrangement is all the more likely to occur that the sequences at the breakpoints are similar. The probability of a chromosomal rearrangement to occur hence depends on the sequence itself and consequently, is subject to evolution. Details about how we modeled these homology-driven chromosomal rearrangements are provided in the next section.

Genetic exchange (crossover) between individuals was not allowed in the simulations presented here, because we first needed to assess the impact of similarity-based intra-chromosomal rearrangements in the simple case of an asexual population. We plan to allow for similarity-based genetic exchange in future experiments.

### Homology-driven chromosomal rearrangements

Taking homologies into account in the chromosomal rearrangement process requires some knowledge regarding sequence repeats on the chromosome. A naive approach would be to compute a complete alignment search of the genome on itself and then to proceed to the rearrangements if any. However, searching for alignments between sequences is known to be a computationally costly problem. In our particular case, where we deal with millions of genomes (classically 1,000 genomes per generation for thousands of generations), even a heuristic search such as BLAST (Altschul et al., 1990) would be forbiddingly long to compute. Another possible approach, chosen here, is to use intermittent searches (Bénichou et al., 2005), that provide us with a partial yet sufficient knowledge of sequence alignments within the genome.

In bacteria, several mechanisms can result in a rearranged chromosome. All these mechanisms have a basic prerequisite of spatial proximity: two sequences must be physically close together in the cytoplasm, at least at the breakpoints, for them to rearrange. As the chromosome is supercoiled, two sequences that are very distant from each other on the chromosome can very well be next to each other in the three-dimensional conformation. Since the mechanisms that constrain the spatial conformation of the genome according to its sequence are still poorly understood in bacteria, here we simply picked random pairs of sequences on the genome and consider them to be neighbours.

How many pairs of points are to be drawn depends on both the genome length and its degree of supercoiling. Consider any given sequence on the genome. The number of other sequences that are localized in its surroundings depends on how densely packed the genome is. In a highly supercoiled genome, for instance, all the sequences are very

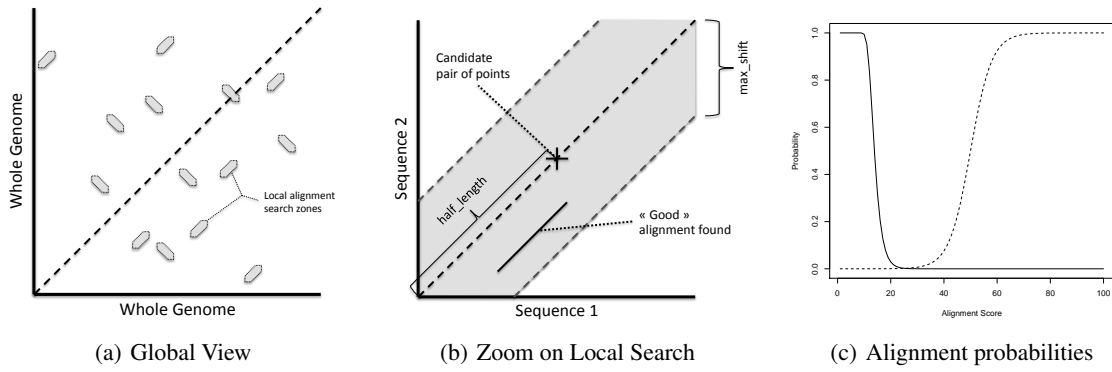


Figure 4: (a) For each pair of points that are candidate for a rearrangement to occur, a local alignment search is performed between the surrounding sequences either in direct or indirect sense. (b) The searching zone is defined by 2 parameters: the *half\_length* of the searching zone and the maximum slippage *max\_shift* authorized between the sequences. In the experiments presented in this paper, we used values of respectively 50 for *half\_length* and 20 for *max\_shift*. (c) Solid line: probability to find a sequence of the given score on a random sequence. Dashed line: the function  $p_{rear}(score)$  used to map scores to rearrangement probabilities in our experiments.

tightly packed together so any sequence has many neighbours and thus many rearrangement opportunities. We thus introduced a specific parameter in the model, the “neighbourhood rate” ( $\mu_n$ ), that expresses this degree of supercoiling. The number of pair of points to consider for a possible rearrangement will then be given by  $L * \mu_n$ , with  $L$ , the genome length in bp. Here,  $\mu_n$  is a parameter defined for the whole population and cannot change during its evolution.

For each candidate pair of points, a basic local alignment search will be performed to determine the existence of similarities between the surrounding sequences either in a direct or indirect sense (figure 4(a)). To that end, we defined a simple scoring function (+1 per match, -2 per mismatch) that allows us to quantify the similarity of two sequences<sup>1</sup>, and associated each score to a probability of rearrangement. The kind and number of rearrangements are computed thanks to algorithm 2.

Preliminary experiments allowed us to adjust the function  $p_{rear}(score)$ , that maps alignment scores to probabilities of rearrangement. To favour homologous over non-homologous rearrangements, alignment scores that are seldom found on random sequences (high scores) are associated with very high rearrangement probabilities (homologous rearrangements). Low score alignments on the other hand, are likely to result from contingency, and will hence be given low probabilities of rearrangement (nonhomologous rearrangements). Figure 4(c) shows the probability of finding an alignment of a given score on a random sequence as well as the function  $p_{rear}(score)$  we used in the following experiments. This particular function yields a reasonable tradeoff between homologous and nonhomologous

rearrangements.

```

initial_nb_pairs ← L * μn
nb_pairs ← initial_nb_pairs
while nb_pairs > 0 do
    Draw 2 random positions pos1 and pos2
    Draw type of rearrangement
    if Inversion then sense ← indirect
    else sense ← direct
    Draw minimal alignment score using prear-1
    Search Alignment(pos1, pos2, sense, min_score)

    if Alignment found then
        Proceed to Rearrangement
        Update L
    end
    nb_pairs ← nb_pairs - 1
    nb_pairs ←  $\frac{nb\_pairs}{initial\_nb\_pairs} * L * \mu_n$ 
end

```

**Algorithm 2:** Aevol Rearrangement Process Algorithm

## Results

Our model being quite complex, our experimental methods are very similar to those used in “wet” experimental evolution. We let 60 populations of 1,000 asexual individuals evolve during 20,000 generations in near identical conditions where the only changing parameters were the mutation rate (one common rate  $\mu_m$  for the three different types of local mutations, 4 values ranging from  $5.10^{-6}$  to  $1.10^{-4}$  were tested) and the neighbourhood rate ( $\mu_n$ , 4 values ranging from  $1.10^{-2}$  to  $5.10^{-1}$ ). During the evolutionary process, the organisms progressively acquire new genes by duplication and modify them in such a way that the whole gene repertoire fulfills the task the organisms are selected for.

<sup>1</sup>Even though it is possible to allow for gaps within alignments, the computation cost would be too important. Hence, in the experiments presented here, no gaps were allowed.



All the simulations proceed qualitatively in a similar way, evolving quickly in the first stage of evolution (rapid gene acquisition mostly by duplication-divergence) then slowing down the process of gene acquisition while optimizing the sequence of existing genes and promoters.

In the experiments presented here, the rate at which rearrangements occur is not constant, it depends on both the neighbourhood rate  $\mu_n$  and on the presence of repeated sequences on the chromosome. It is hence free to evolve and could well be selected for or against. Yet, despite this added degree of freedom, the rearrangement rate remains a very strong determinant of genome size and content (figure 5). These results confirm those obtained with previous versions of the model in which the rearrangement rates were direct parameters of the model (Knibbe et al., 2007). Even with homologous rearrangements, we find again that the spontaneous rate of rearrangement has a negative impact on fitness (figure 5(d)) because it sets an upper bound on genome size and hence on the number of genes (figure 5(c)). However, rearrangements are also mandatory for evolution to be efficient. An organism whose genome would have lost its capacity to rearrange would hardly be evolvable at all.

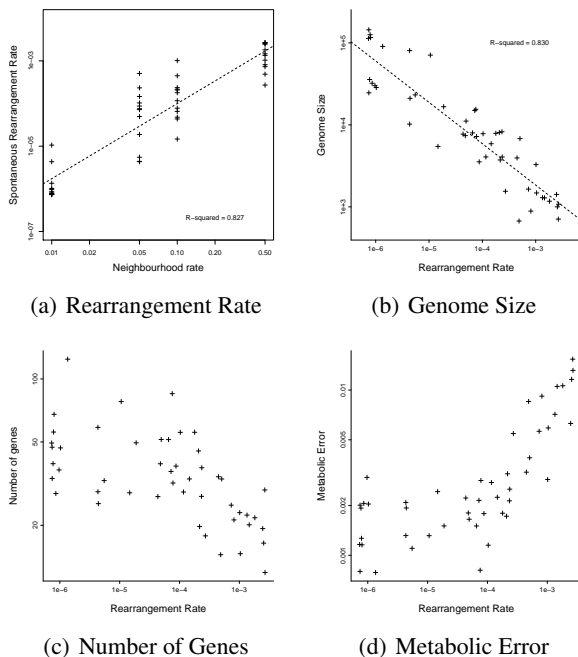


Figure 5: (a) Average spontaneous rearrangement rates observed for each simulation during the whole evolution. (b,c,d) Genome Size, Genes Number and Metabolic Error of the best organism after 20,000 generations for each simulation, as a function of the spontaneous rearrangement rate.

Because homologies are created by rearrangements (duplications) and gradually destroyed by local mutations, there must be some sort of complex interactions between the mutation rate, the neighbourhood rate and the rates of both ho-

mologous and nonhomologous rearrangements.

The distribution of the scores of the alignments that led to rearrangements (figure 6) can help us understand this intricate relationship. If we consider this data vertically, we can clearly observe that the proportion of homologous rearrangements is higher when the neighbourhood rate is high. However, as we progress downwards, the distributions behave differently: while it remains nearly unchanged on the left hand side, nonhomologous rearrangements become way more frequent on the right. A noteworthy observation is that there is a great variation in the number of rearrangement events. In fact, it is not the number of nonhomologous rearrangements that raises (it actually remains stable), but rather the number of homologous rearrangements that collapses when the neighbourhood rate decreases.

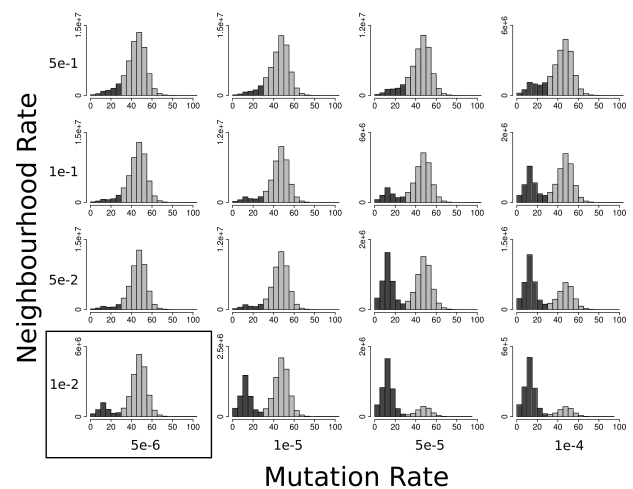


Figure 6: Distribution of the scores of the alignments that caused a rearrangement to occur in the whole population and during the entire evolutionary process, for each value of  $\mu_n$  and  $\mu_m$ . Light grey: homologous rearrangements, dark grey: nonhomologous rearrangements. For computational performance reasons, the given values are minimal bounds to the corresponding alignment score (cf. Algorithm 2).

The underlying phenomenon is best understood when looking at the data in a top-left to bottom-right fashion. One can then identify a phase transition between a regime of mainly homologous rearrangements at high  $\mu_n$  and low  $\mu_m$ , and a regime of almost exclusively nonhomologous rearrangements at low  $\mu_n$  and high  $\mu_m$ . In fact, for the possibility of homologous rearrangements to be maintained along the evolutionary process, homologies must be created (by either homologous or nonhomologous duplications) at least as fast as they are destroyed by local mutations. At high neighbourhood rates, this condition is always achieved because rearrangements are numerous. However, at low neighbourhood rates, the damage caused by local mutations can overcome the creation of homologies and stall the whole process.

The four histograms at the bottom of Figure 6 are hence the most interesting. Within this line, throughout which  $\mu_n = 1.10^{-2}$ , the change in rearrangement mode from mainly nonhomologous to mainly homologous is particularly clear when the spontaneous rate of small mutations decreases. To better understand the dynamics of homologous/nonhomologous rearrangements, we further analysed the simulations from the left hand side, that display both the greatest proportion of homologous rearrangements (within the bottom line) and, interestingly, the best final fitness of all parameter sets. For the three runs of this parameter set ( $\mu_n = 1.10^{-2}$  and  $\mu_m = 5.10^{-6}$ ), we kept track of the family ties during the evolution. We then retrieved the line of ancestry of the final best individual and analyzed the mutational events that occurred on this successful lineage. Except for those that occurred during the very last generations, the events on this lineage are those that went to fixation, either by selection or by genetic drift. In addition, every other 10 generations, we used the standard bioinformatic tool Mummer (Kurtz et al., 2004) to find the most significant repeated sequences in the ancestral genome. Mummer uses an approach similar to that of BLAST, it first searches for exact short repeats and then tries to join them together, allowing for gaps and mismatches. An example of Mummer output is shown in Figure 7. In this example, there are both direct and inverted repeats, and most of the repeated sequences are located in non-coding parts of the genome. This suggests that non coding DNA plays a major role in genome evolvability by providing breakpoints for chromosomal rearrangements. The emergence of repeated sequences having little or no direct impact on fitness has already been observed in genetic programming (Langdon and Banzhaf, 2008) though in that particular case, these repeated sequences could be thought to participate in robustness rather than evolvability.

Figure 8 shows the results of the analysis of the whole lineage of ancestors. It shows that fitness improvements are strongly correlated with the presence of repeats in the genome and, consequently, with the occurrence of chromosomal rearrangements. The impact of chromosomal rearrangements on evolvability is thus rather complex: on the one hand, a very high rate of spontaneous rearrangements has a negative impact on the final fitness (Figure 5(d)), but on the other hand, in these simulations where the rate was low and the final fitness high, we find that the presence of rearrangements is correlated with fitness improvement (Figure 8). This suggests that a minimal amount of chromosomal rearrangements is required for evolution to be efficient.

A closer look to the rearrangements that went to fixation in these simulations (see Figure 9) reveals that (i) most of the fixed rearrangements were based on homologous breakpoints (*score* > 40), (ii) most of the fixed translocations and inversions were neutral, (iii) most of the fixed deletions were beneficial and (iv) most of the fixed duplications were deleterious. This last result is surprising at first sight: one would

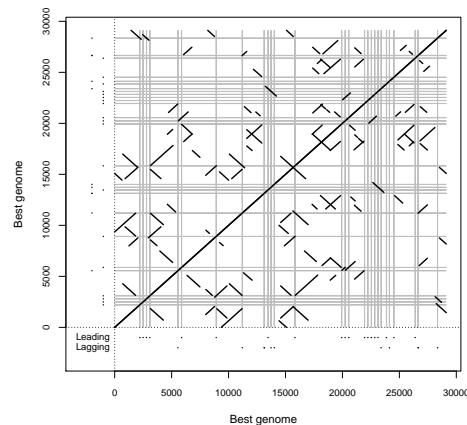


Figure 7: Example of Mummer “dot plot” for the best individual at  $t = 2000$  generations, for  $\mu_n = 10^{-2}$  and  $\mu_m = 5.10^{-6}$ , seed 2. Both the x- and the y-axis represent the genome of this individual. Long and strongly similar sequences appear as runs of diagonal lines across the matrix (exact match length = 15 bp, min. cluster length = 200 bp, max. gap between adjacent matches = 6 bp). Grey areas: coding sequences.

expect fixed events to be mostly neutral or beneficial. Our hypothesis is that despite their immediate negative impact, duplications can be indirectly selected because they allow for the creation of new gene copies (which can then undergo small mutations and ultimately realize new functions) and new repeats (which can then mediate other rearrangements).

## Conclusion

These experiments of *in silico* evolution with similarity-based rearrangements confirm our previous results regarding the influence of rearrangements on genome compactness. In large genomes, repeated sequences (located mostly in non-coding regions) promote rearrangements that are, most of the time, deleterious. There is thus an indirect selective pressure to limit the number of rearrangements, which is done by eliminating repeats (fewer homologous rearrangements) and by reducing genome size (fewer nonhomologous rearrangements). However, we have also shown that the absence of rearrangements is correlated with fitness stasis, suggesting that rearrangements can sometimes be directly beneficial or provide appropriate genetic background for subsequent beneficial mutations. A minimal amount of rearrangements is thus required for evolvability. Here, most of the rearrangement kept by evolution are homologous ones. For them to be possible, repeats must be created at least as fast as they are destroyed by small mutations. In the end, the best conditions for evolvability seem to be a small basal rate of nonhomologous rearrangement combined with a low-enough mutation

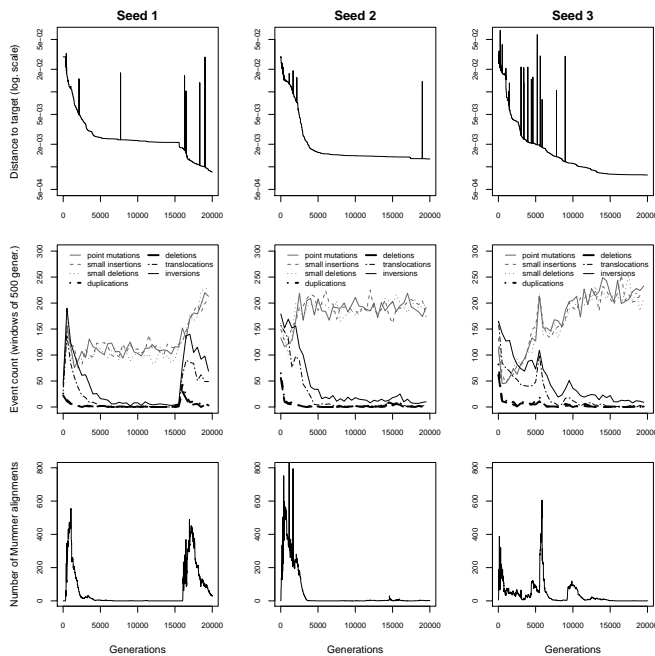


Figure 8: Analysis of the line of ancestry of the final best individual for  $\mu_n = 10^{-2}$  and  $\mu_m = 5.10^{-6}$ . First row: evolution of the fitness (the smaller the distance to the target, the higher the probability of reproduction). Second row: evolution of the number of mutational events, by windows of 500 generations. Third row: number of alignments found by Mummer on the genome (parameters: see Figure 7).

rate, thus leading to a few stable repeats and to an intermediate degree of variability by homologous rearrangements.

## Acknowledgements

We gratefully acknowledge support from the CNRS/IN2P3 Computing Center (Lyon/Villeurbanne - France), for providing a significant amount of the computing resources needed for this work.

## References

- Adami, C. (2006). Digital genetics: unravelling the genetic basis of evolution. *Nat. Rev. Genet.*, 7(2):109–118.
- Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410.
- Bénichou, O., Coppey, M., Moreau, M., Suet, P. H., and Voituriez, R. (2005). Optimal Search Strategies for Hidden Targets. *Physical Review Letters*, 94(19):198101+.
- Beslon, G., Parsons, D., Sanchez-Dehesa, Y., Pena, J., and Knibbe, C. (2010). Scaling laws in bacterial genomes: A side-effect of selection of mutational robustness. *BioSystems*, 102(1):32–40.
- Blickle, T. and Thiele, L. (1996). A comparison of selection schemes used in evolutionary algorithms. *Evol. Comput.*, 4(4):361–394.

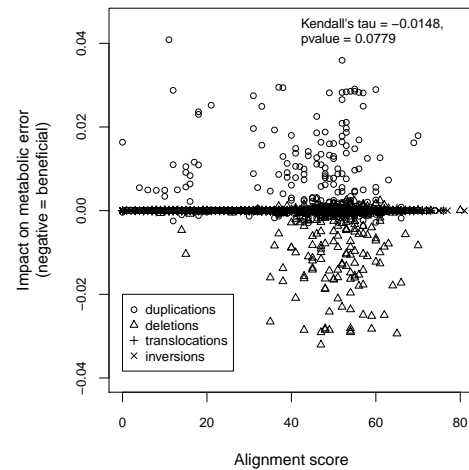


Figure 9: Analysis of the fixed rearrangements for  $\mu_n = 10^{-2}$  and  $\mu_m = 5.10^{-6}$  (all seeds together). Each point represents a rearrangement that occurred on the line of ancestry of the final best individual.

- Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial chemistries-a review. *Artif Life*, 7(3):225–275.
- Earl, D. J. and Deem, M. W. (2004). Evolvability is a selectable trait. *Proceedings of the National Academy of Sciences of the United States of America*, 101(32):11531–11536.
- Kirschner, M. and Gerhart, J. (1998). Evolvability. *Proceedings of the National Academy of Sciences*, 95(15):8420–8427.
- Knibbe, C., Coulon, A., Mazet, O., Fayard, J.-M., and Beslon, G. (2007). A long-term evolutionary pressure on the amount of noncoding DNA. *Mol. Biol. Evol.*, 24(10):2344–2353.
- Knibbe, C., Fayard, J.-M., and Beslon, G. (2008). The topology of the protein network influences the dynamics of gene order: from systems biology to a systemic understanding of evolution. *Artificial Life*, 14(1):149–156.
- Kurtz, S., Phillippy, A., Delcher, A., Smoot, M., Shumway, M., Antonescu, C., and Salzberg, S. (2004). Versatile and open software for comparing large genomes. *Genome Biology*, 5:R12.
- Langdon, W. and Banzhaf, W. (2008). Repeated patterns in genetic programming. *Natural Computing*, 7(4):589–613.
- Misevic, D., Ofria, C., and Lenski, R. E. (2006). Sexual reproduction reshapes the genetic architecture of digital organisms. *Proc. R. Soc. B.*, 273(1585):457–464.
- Neidhardt, F. C. (1996). *Escherichia coli and salmonella: cellular and molecular biology*. ASM Press.
- Parsons, D. P., Knibbe, C., and Beslon, G. (2010). Importance of the rearrangement rates on the organization of transcription. In *Proceedings of Artificial Life XII*, pages 479–486.
- Wilke, C. O., Wang, J. L., Ofria, C., Lenski, R. E., and Adami, C. (2001). Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, 412(6844):331–333.

# A Checkpoint-Orientated Model to Simulate Unconstrained Proliferation of Cells

Jonathan Pascalie<sup>1,2</sup>, Valérie Lobjois<sup>2</sup>, Hervé Luga<sup>1</sup>, Bernard Ducommun<sup>2,3</sup> and Yves Duthen<sup>1</sup>

<sup>1</sup>CNRS - IRIT-UMR5505 - University of Toulouse, France  
{pascalie;luga;duthen}@irit.fr

<sup>2</sup>CNRS - ITAV-UMS3039 - University of Toulouse, France  
{valerie.lobjois;bernard.ducommun}@itav-recherche.fr

<sup>3</sup>CHU de Toulouse - F-31059 Toulouse, France

## Abstract

In this paper we propose a new computational model of cell cycle to study the dynamics of cells population in 2-D monolayer culture. Whereas most of the models are phase-orientated our model deals with a checkpoint orientated paradigm and uses the phase orientation as an output to provide the biologists with a relevant view of the simulation result. Through this paper we will present the genericity of our model, able to reproduce the exponential growth phase of different cellular processes.

## Introduction

Exploring, designing, understanding the complexity of the living world is of tremendous importance. The accurate assessment of its malfunction, especially those related to human diseases is a high stake venture. *In silico* simulation provides new means of studying and exploring living systems. In complementarity with experiments or when they are difficult to address *in vitro*, virtual environments can prove to be of interest. The latest computation capacity explosion allows us to tackle these questions with new approaches and new methods. System modelling may therefore use fitted methodologies to represent living systems at a systemic level. To this aim, the bottom-up approach tends to be the general paradigm for system modelling, focusing on each functional component of the system and in their interactions.

Cancer is often considered as the result of perturbation in cell cycle regulation associated with mutations that can appear in key regulators that result in abnormal proliferation, leading to tumorigenesis. Increasing the understanding of the cell cycle control is therefore central in cancer research and there are high issues in finding new regulatory mechanisms. The pharmacological issues foreseen with the *in silico* simulation of cellular systems let think that prospective research of new therapies could be addressed *in silico*.

In the different fields of computational and molecular biology, the focus on aspects of the cell cycle differs. Molecular biology models focus on the modelling and simulation of the molecular regulatory network of cycline-dependent kinase (CDK) (Novak and Tyson, 2004). These models can

be classified into two kinds of models, the discrete and the continuous. Continuous models basically describe the evolution of concentration of proteins using a set of ordinary differential equations, whereas discrete models focus on the activation state of each regulatory protein thanks to a predefined genetic regulatory network (GRNs) (Kauffman, 1969; Chavoya and Duthen, 2008). These models have been commonly used to simulate the cell cycle in yeast (Chen et al., 2004; Novak et al., 2001), frog eggs (Novak and Tyson, 1993; Pomerening et al., 2005), fruit flies (Calzone et al., 2007) and different mammals cells (Aguda and Tang, 1999; Singhania et al., 2011). These models are molecular-based models and do not account for behavioural considerations at a macro-level, their aims being to focus on the regulatory mechanisms.

The other family of models used to simulate cell proliferation are called Individual Cell-Based Models (IBMs) (Loeffler and Roeder, 2004). These are a subset of the agent-based models. Agent-based models have mainly proved their relevance in the simulation of different complex systems from social networks to the social behaviour of hive insects. Basically, individual cell based models come under two classes: cellular automaton (CA) models and off lattice models. On the one hand, CA are described by a discretization of the proliferative environment in 2-D/3-D evolution grid, and the cell shape is reduced to a lattice site. In this case, cell behaviour is composed of the different update rules set up (Patel et al., 2001; Moreira and Deutsch, 2002). On the other hand, off-lattice models have the advantages of letting evolving cells in a continuous media with continuous shapes. They can introduce topological aspects based on *in vitro* observation or knowledge. This involves high stakes for some investigative considerations. The IBMs have been successfully used to study the pattern formation in multicellular cultures (Galle et al., 2005; Gerlee and Anderson, 2007), avascular tumour growth (Hoehme and Drasdo, 2010) and the spatio-temporal organisation of tissues (Meineke et al., 2001; Drasdo and Loeffler, 2001). These models generally consider the cell cycle as a single time unit decision and the update frequency is the global scheduler of the cell cycle.

Basically, this representation does not allow any consideration on the cell cycle phases.

Moreover, IBMs and hybrid representations with GRNs have been widely used in Artificial Life to study the mechanisms of morphogenesis (Cussat-Blanc et al., 2010; Doursat, 2006). In these studies the cell cycle has to be seen as the cell behaviour with a bio-inspiration paradigm.

Whereas molecular-based models well express the dynamics of advancement of cells in each phase of the cell cycle, the individual-based models often do not, due to their description of the cell cycle. Expressing these dynamics reveals interest to simulate some *in vitro* culture where external compounds are introduced to study their effect in the dynamics of advancement. In this work, our goal is to simulate as closely as possible the population response to an external stress expressing the dynamics of progression of the cells at a population scale. For that purpose, we use the simplicity of IBM representations to describe the cellular behaviour and to introduce temporal considerations thanks to an accurate description of the cell cycle. This approach leads us to build a hybrid representation of the cell cycle with a hand coded regulation network and probabilistic-based cellular processes. In this paper we will show the preliminary results obtained simulating a particular stage of the cell's population dynamics: the exponential growth phase, allowing us to focus on the population dynamics for the sequencing of the different phases.

The following sub-section presents the biological background of this study. Section 2 extends the model proposed in (Pascalie et al., 2010), presenting its computational aspects with a formal representation. Section 3 shows preliminary results of cell proliferation in an exponential growth phase. Finally, the last section concludes and discusses the problem of parameters tuning based on the results presented in section 3.

## Biological Background

The cell cycle is often drawn as a circular timeline with different phases starting in G1 and ending at mitosis when a cell divides into two daughter cells. The study of the cell cycle by the biologists puts major emphasis on the essential role of the checkpoints (Elledge, 1996). They are the warrants of the cell's genomic stability and their integrity ensures a good progression on the cell cycle timeline. By the end of the G1-phase, at the commitment point (R), the cell integrates environmental signals before proceeding towards the G1/S transition. A lack of these signals will lead the cell to enter a quiescent (G0) state. If pro-apoptotic signals are detected the cell will undergo death, called apoptosis. Alternatively, differentiation signals will drive the cell out of the cell cycle to a differentiation programme. If the cell progresses in the cell cycle, it must duplicate accurately all its internal material (DNA, centrosome etc) and double its mass before preparing for division. Before entering into

S-Phase where DNA synthesis occurs, the cell must check for the integrity of its genetic material. This is called the G1/S DNA integrity checkpoint. Providing that DNA synthesis is fully completed, the cell switches to G2-phase and it finishes doubling its mass. During S-phase and G2-phase, centrosome duplication and maturation occurs thus building the two platforms that will allow the assembly of the mitotic spindle required for mitosis to occur. However, before proceeding from G2 to mitosis, the cell must ensure the integrity of its genetic material again. This is called the G2/M checkpoint. At mitosis, when cells are dividing, in order to ensure even segregation of the genetic material in the two daughter cells, the mitotic checkpoint (iM) prevents division until the chromosomes are perfectly aligned on the equatorial plan. Any alteration in these checkpoint mechanisms (for instance a mutation in a key regulator) leads to a genetic instability often associated with transformation and cancer. For these reasons it is essential to integrate checkpoints as artefacts (or essential milestones) of our simulation model. Figure 1 shows a cartography of the cell cycle with the localisation of each cellular processes and checkpoint.

## Modelling Cell Behaviour

This work focuses on the temporal behaviour of cells and the different regulatory mechanisms (i.e the checkpoints) are emphasized to study their influence over population scale. This problematic drives the modelling process. For that purpose, the cell cycle specificities are described and embedded in our representation as closely as possible to *in vitro* cell cycle.

The study of the cell cycle points out the cleavage between the functional level and the regulation level. Simulation models often focus on one of these aspects, however the effective cell behaviour depends on the interaction between these two levels. In fact, the changes made on the cell's internal state by the functional level (e.g. doubling the DNA)

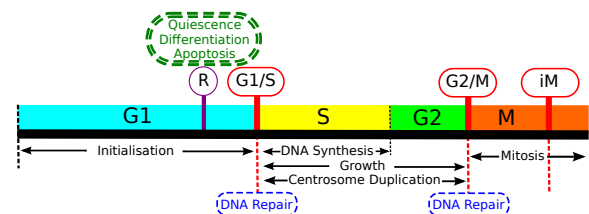


Figure 1: Localization of different cellular processes and checkpoints on the cell cycle timeline. Red simple-lined boxes represent checkpoints with iM being the intra-mitotic one; blue dotted boxes are processes that could be executed during the associated checkpoint; in black with arrows are represented the different processes executed during the cell cycle; the ringed R is the commitment point, another regulator, and the green double-dotted box represents the three exiting points

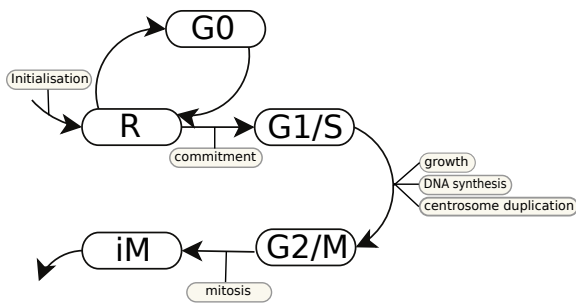


Figure 2: The defined regulators are connected to each other to build a finite state machine (FSM) which embed the regulatory mechanisms of the cell cycle. The schema also indicates in which position of the FSM are executed each processes.

drive the cell on a specific regulatory pathway. To represent these mechanisms and their interaction with the best accuracy possible it is necessary to observe and describe both levels in a cell cycle accurate modelling.

The two next parts will first present how we designed the cellular behaviour and its computational implementation.

## Cell Cycle Instance

The behaviour of the cell described in the introduction induces a split in sub-behaviours which represents specific cellular processes. The good sequencing and scheduling of these processes is therefore managed by the different checkpoints and/or regulators clarified above.

Figure 2 shows the network of regulators we have defined to manage the cell behaviour at a high level. The cell starts its lifecycle with as first goal to try to pass the restriction point (R). The cell cycle is therefore defined by the  $R \Rightarrow G1/S \Rightarrow G2/M \Rightarrow iM$  sequence of regulators. This pathway ends with the mitosis of the cell and represents the proliferative behaviour of the cell.

This modelling approach allows us to build a generic cell cycle model which could be used to design specific cells by instantiating specific checkpoints and processes. The following list describes the different cellular processes we designed in this study. These cellular processes have to be seen as the behaviour of the cells *during* the transition between two nodes:

- **Initialisation:** this action matches the G1-phase of the cell cycle. All cells starting their cell cycle observe this phase, which culminates at the R restriction point. During this phase, the cells have not yet been committed to proliferation, differentiation nor entry into quiescence. This process is more a delay prior to commitment point (R) than a cellular process.
- **Commitment:** this action is the planning behaviour of the cell. It occurs when the cell has ended its initialisation

and when it decides which behaviour it will execute.

- **DNA Synthesis:** this activity represents the S-phase of the classical cell cycle. It starts at the end of DNA repair - if necessary - when DNA integrity has been verified at the G1/S transition. During this action the cell replicates its DNA.
- **Growth:** this action represents the cell's mass doubling. It starts at the beginning of the S-phase and ends during the G2-phase.
- **Centrosome Duplication:** this action represents the duplication of the centrosome. It occurs simultaneously with Growth during the S- and G2-phases.
- **Mitosis:** it is the last action of the cell cycle. It requires prior checking of genomic activity at the G2/M transition. If all pre-conditions are met, mitosis occurs in the final stage of the cycle and ends with the beginning of the two new cycles of the daughter cells. Completion of mitosis requires chromosome alignment at the equatorial plan (mitotic checkpoint).

A cell is thus considered to be in G1 phase until it has passed the G1/S checkpoint (if it is executing *initialisation* or *commitment* activities to be precise). A cell is considered in the S-phase while it is executing *DNA synthesis* regardless of *growth* and *centrosome doubling*. Therefore the cell is considered in the G2 phase when it has ended its DNA synthesis and while it is ending its *growth* and its *centrosome doubling*.

The proliferation is not the only behaviour observable in this model. The regulatory network presents alternative behavioural functions of the pathway followed by a cell:

- **Differentiation** represents one of the exit points of the cell cycle. If specific conditions are met the cell will differentiate. This exiting point is available at the R node (Restriction Point) of the regulatory network.
- **Quiescence**, also named G0-Phase, is an active survey loop used when environmental factors are insufficient for the cell proliferation. The quiescent cells are able to return to the cell cycle at any time if the growing conditions are met. This alternative behaviour occurs when the cell is at the G0 node.
- **Apoptosis** represents cellular death. Apoptosis happens if apoptotic factors or signals are delivered to the cell or if the cell spends too much time in a specific stationary situation of its cell cycle. Apoptosis can occur at any time of the cell cycle.

To process the cell behaviour, the regulators (i.e the nodes of the network) are composed of a list of activities along with the preconditions of their activation. The regulators are

global schedulers for the cell cycle and have the same role as the checkpoints in real cells. They regulate the cell cycle and activate the different processes if their preconditions are fulfilled. If several activities are activated at the same time the cell executes them simultaneously. The preconditions are two sets of boolean flags:

- one representing the internal state of the cell or its disponibility of environmental factors,
- the other indicating which activities are done; under progress or planned.

The following list presents the different regulators we defined in our computational cell cycle model:

- The **R commitment point**: cell has to choose between commitment to the proliferation pathway, the quiescent stage, or the differentiation process.
- The **G1/S checkpoint**: here the cell checks its DNA for lesions. If lesions are found, the cell repairs them or else it starts DNA Synthesis, Growth and Centrosome cycle.
- The **G2/M checkpoint**: to pass through this checkpoint the cell must have replicated its DNA, should not have detected any DNA damage, have duplicated its centrosome and doubled its mass.
- The **intra-mitotic checkpoint**: to pass this checkpoint and to divide into two daughter cells, the cell needs to have aligned its chromosomes on the mitotic plan and placed its centrosomes on the mitotic spindle poles.
- The **G0 regulator**: we choose to model the G0 state as a regulator because it represents an active survey loop of environmental factors for proliferation. In order to uncorrelate the cell functional level and its regulation, we consider this particular state as a regulatory element of our cell cycle model.

DNA repair has also been added as an activity to include and study the influence of the timings of DNA damages repair.

## Computational Model

A natural population of cells presents heterogeneous features. Owing to the variability of the duration of each cell cycle phase, two cells born at the same time will not divide simultaneously even if environmental conditions were equivalent. This property is extracted from *in vitro* cultures. To represent this heterogeneity we choose to let the parameters embedded in each cell be generated according to a distribution law. Our cell cycle model is thus defined to produce a population of a specific cell type and not a single cell. If the population of cells was only represented as a population of clones of a given cell, the dynamics of the cell population

would suffer from phasing in the sequencing of the different phases, each sister cell going to division at the same time.

To represent the cellular activity in a temporal manner and remain at a macroscopic level of representation<sup>1</sup>, we based the cellular process modelling on their scheduling. In this context, 3 parameters are used for each cellular process  $i$ : the optimal time of realisation, the maximum time before it eventually results in cell death, and the probability of success. A cellular process  $i$  is thus modelled using the following parameters:

- the average optimal time of a process  $i$ :  $\mu_{avg}^i \in \mathbb{R}^+$
- the standard deviation at the population scale of the average optimal time:  $\sigma_{avg}^i \in \mathbb{R}^+$
- the average maximal time of a process  $i$ :  $\mu_{max}^i \in \mathbb{R}^+$
- the standard deviation for the average maximal time:  $\sigma_{max}^i \in \mathbb{R}^+$
- the probability of success for the process  $i$ ,  $P_s^i \in [0; 1]$ , which has to be interpreted as an efficiency potential.

To integrate the population heterogeneity the previous parameters are defined for a population. Using these parameters, we generate a set of parameters which are used for the computation. Our processes are represented over time as Bernoulli processes. The average optimal time determines the number of successes needed to consider the process as achieved. The success rate is used to define the probability of success of one trial.

We can then specify for a given cell and a specific process  $i$ :

- the optimal time to finish the process  $i$ :

$$T_{opt}^i \sim \mathcal{N} < \mu_{avg}^i, \sigma_{avg}^i >$$

- the maximal time to finish the process  $i$  before death:

$$T_{max}^i \sim \mathcal{N} < \mu_{max}^i, \sigma_{max}^i >$$

These parameters are chosen to vary over the population thanks to a normal law that follows the probability density function with  $\mu$  as mean and  $\sigma$  as standard deviation:

$$\forall t \in \mathbb{R}, \quad \phi(t, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}$$

This definition does not allow a zero valued  $\sigma$ . To ensure the availability of building simulation without heterogeneity, a standard deviation set up to 0 affects the specified  $\mu$  as the value of the parameter for all the cells of the population. This kind of parametrisation could be used to compare

<sup>1</sup>we do not want to model molecular interaction nor the genetic regulatory network yet



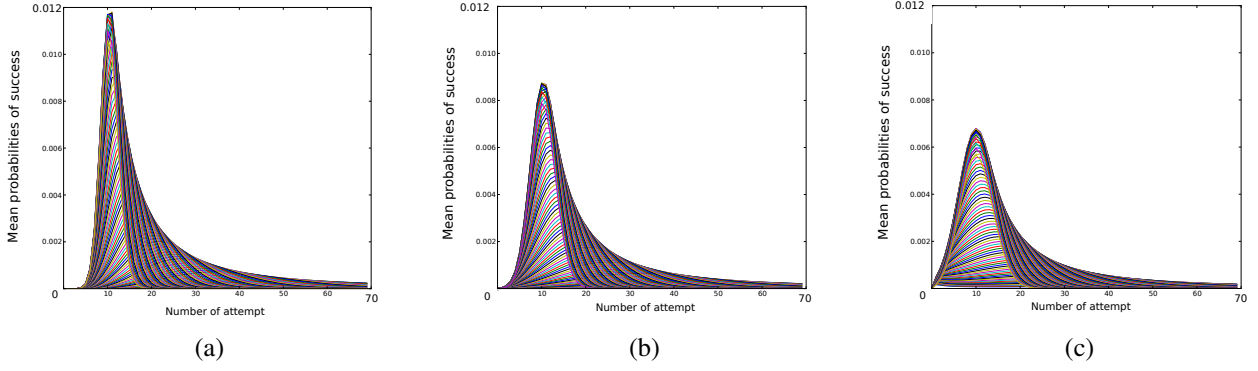


Figure 3: Example of the configuration of mean probabilities for a given process with variation of the probability  $p$  with different values of the standard deviation  $\sigma$ . The mean value for the process is defined with  $\mu = 10$ . (a)  $\sigma = 1.5$ . (b)  $\sigma = 3.5$ . (c)  $\sigma = 6.5$

the influence of the different parameters or to study specific abstract behaviours.

These generated parameters help us to customise the Bernoulli process, which takes place to compute cellular activity. Parameters that have to be defined to map the cell model on the computational model are the frequency of the Bernoulli experiences over time. This frequency will allow us to have an accurate temporal model of cells behaviours, allowing the modeller to express heterogeneity regardless of the duration of the cellular process. The last parameter that needs to be defined to end this mapping is the duration of a simulation step. This has to be done during the setup of a specific simulation and not during the cell modelling because the cell model must be as scalable as possible beside the simulation step. The Bernoulli process  $i$  taking place to simulate a cell behaviour is parametrised as follows:

- the number of successes a cell has to reach to consider its process  $i$  as well terminated:  $n_s^i \in \mathbb{N}$ ,
- the probability of success for the bernoulli experience of the process  $i$ :  $P_s^i \in [0; 1]$ ,
- the maximum number of attempts to reach  $n_s^i$  successes:  $n_{max}^i \in \mathbb{N}$ .

To help the biologist in cell cycle modelling it is important to provide him with a representative view of the parameter's influence. In this case, formal representation can help us to give a relevant view of the initialisation population. As the temporality of the cell's activities is at the centre of the modelling process, the mean probability of success over time is defined so as to characterise the dynamics of the population over time.

The probability of success in a specific process  $i$ , at the  $k$ -th trial with the probability  $p$ , for a specific cell is given by the

negative binomial law:

$$\forall k \in \mathbb{N}, \forall p \in ]0; 1[ \\ \Lambda(k; n = T_{max}^i, p = P_s^i) = C_{n+k-1}^{n-1} \cdot p^n \cdot (1-p)^k$$

This law models the probability of failure (*resp.* success) that can appear before reaching a number of successes (*resp.* failures) defined by  $n$  with each attempt associated with a probability of success  $p$ . More generally the probability of success of a process  $i$  for a given cell is given by the repartition function applying to the maximum number of attempts  $T_{max}^i$ :

$$P_s(k = T_{max}^i) = 1 - q^{k+1} \sum_{i=0}^{n-1} C_{K+i}^i \cdot p^i$$

The generalisation of the previous equations at a population scale needs to take into account the distribution of the parameters of the binomial law. The following equation computes the mean probability of success of the process  $i$  at a given Bernoulli trial  $k$  with  $\mu$ :

$$\forall k \in \mathbb{N}; \forall \mu, \sigma \in \mathbb{R}^+; \forall p \in ]0; 1[ \\ \Phi(k, \mu, \sigma, p) = \frac{1}{k} \sum_{i=0}^k P(i, \mu, \sigma) * \Lambda(k-i; i, p)$$

Figure 3 shows the interpretation of the previous equation. It can be observed that a high granularity of representation for the different cellular processes is offered to the biologists. This granularity gives to the model its genericity properties. The next section presents the preliminary results obtained with an abstract model of cell cycle based on *in vitro* observations.

## Simulation of Cells in exponential growth phase

In this paper we will study the dynamics of the cell population in an exponential growth phase. In this part we will take

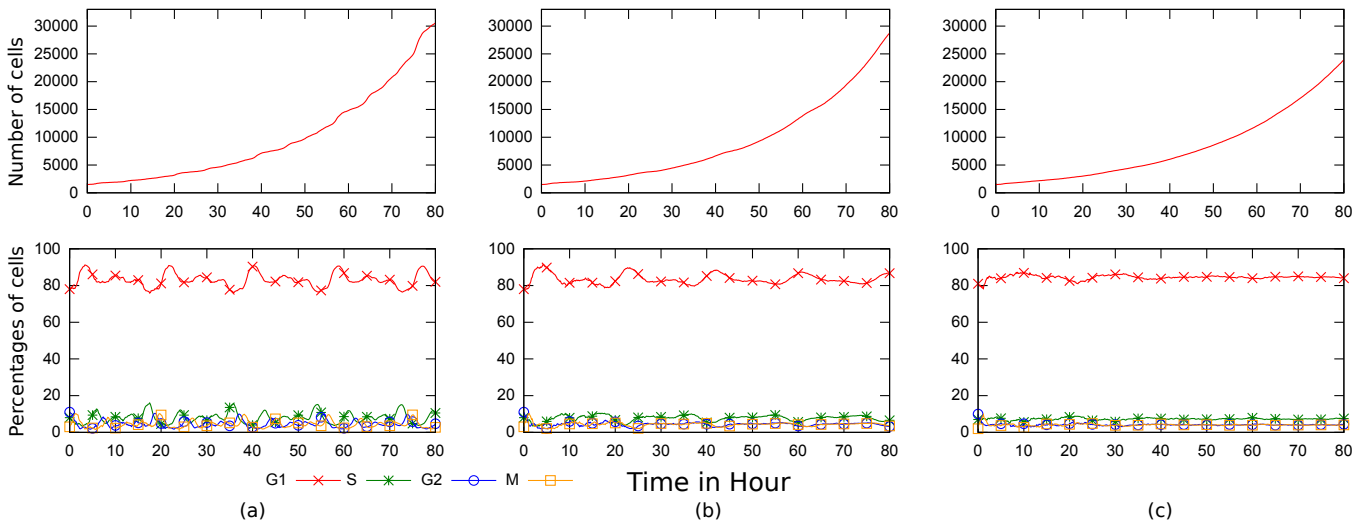


Figure 4: example of unconstrained cell development for different variability. In (a), figure shows that the system is phasing because of a too weak standard deviation over the population. Figure (b) shows the phasing pattern attenuating itself thanks to a higher variability on the G1 phase. Increasing more the standard deviation induces a flattening of the curve and a constant evolution in each phase over time which matches with *in vivo* properties.

advantage of specific features of cell proliferation that allow strong simplification of the environment. Indeed, during exponential growth phase, cell proliferation is not inhibited by environmental signals, the environment being saturated in growth factors. This property allows the simplification of the chemical aspects of the environment, dispensing it with diffusion algorithm and chemical reactions.

The other inhibition factor undergone by the cells is the contact inhibition. It is not observed, in an exponential growth phase. This specificity allows us to dispense the environment and the simulation with physical consideration. The cells do not have shapes and they do not need to interact with each other. As our goal is to study the dynamics of the population and not its topological aspects, these simplifications are adapted to this first step.

In an exponential growth phase, the cell proliferation reveals that the ratio of cells in each phase (G1,S,G2,M) of the cell cycle remains constant over time. This property must be expressed in our model before testing the immersion of cells in a complex environment. To test this, an abstract model of cell behaviour based on the HCT116 cell (a colon cancer cell line often used by biologists for *in vitro* studies) lineage was designed. Experimentally, we have determined, by flow cytometry analysis, that in *in vitro* culture conditions, these cells spend 81% of their cycle in G1-phase, 10% in S-phase, 7% in G2-phase and 2% in Mitosis, for a global duration of the cell cycle of 18 hours. These measures are used to set up the  $T_{opt}$  value of the different activities.

These percentages also correspond to the distribution of cells in the different phases of the cell cycle. These param-

eters determine the initial distribution of the cells for the simulation. It should remain constant over time if the cell model is generic enough and if the designed population is heterogeneous enough.

Figure 4 shows exponential growth simulations executed with different values for the standard deviation  $\sigma_{max}$ . Those results are the average of 10 runs with the same parameters. It is important to specify that even though the random generator seed is not the same from one run to another the stability of the simulation results is not affected.

Curves (a) show the dynamics of the population with an homogeneous population of cells. We can observe that the dynamics of the population oscillates between each phase and that the initialisation pattern reproduces itself over time.

Curves (b) show a simulation where the variability of the *initialisation* action and of the *commitment* action have been increased. The initialization pattern is still present but we can observe that it is attenuating over time. This is essentially due to the fact that, after a cell division, all temporal parameters are reinitialise in daughter cells.

Curves (c) show the result of a 2-D culture with a high variability of *initialisation* and *commitment* action. The evolution of the population in each phase is constant over time. The heterogeneity of the population is high enough to maintain a constant rate in each phase.

In order to evaluate the heterogeneity of the population, the second experiment consists in taking only cells in mitosis from a population and to observe the desynchronisation of that population over time. Figure 5 shows, with the same parameters as in the previous experiment, the development

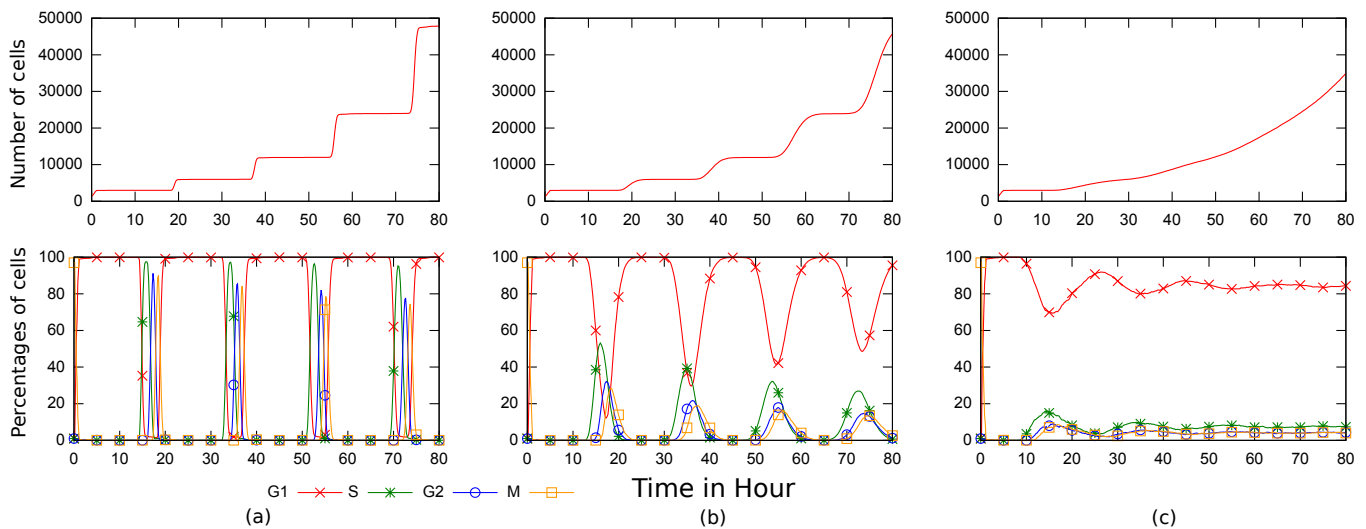


Figure 5: Example of population unsynchronisation, we can observe on the population growth curve the different steps before reaching a power law, if the simulation had lasted longer the dynamics of the population would have remained constant. (a) No unsynchronisation is observed because of homogeneous population. (b) and (c) The attenuation pattern is function of the degree of variability introduced in the simulation.

of a 2-D culture with all the initial population synchronised in the mitosis phase. The cells are not synchronised in a temporal manner, this means that they will not divide at the same time except in the first population which is homogeneous. As previously, the experiment (a) with fixed temporal value does not unsynchronise over time. The second experiment (b) shows that the population unsynchronises over time but that the phasing pattern needs more than 7 or 8 cycles to unsynchronise whereas in the third experiment (c) it loses synchrony by the end of the fourth cycle. More generally, if the cell population is heterogeneous enough, whatever the parameters of initialisation are, the dynamics of the population will balance itself until it observes a constant evolution over time.

## Conclusion

These preliminary results point out that the model can reproduce the dynamics of cell proliferation in an exponential growth phase. What becomes apparent is that the model - and more globally the simulations - is difficult to tune because of the high number of parameters. The main difficulty of tuning is related to undimensional parameters that cannot be defined with values measured with *in vivo* experiments. In the proposed model there are two kinds of parameter sets. The first one is the set of the known biological parameters. It is the set of the parameters that can easily be tuned thanks to biological knowledge or measurement. The other set contains some control parameters, which are difficult to tune because they define a global dynamics for the system. The influence of the different values is not predictable, therefore predicting the influence of the different combinations

becomes difficult.

We propose a strategy to validate the cell cycle models before it. The *in vitro* observation of cells exergues that the ratio of each cell phase during an exponential growth is constant. With this piece of information, the ratio of cells in each phase and the global duration of the cell cycle, the optimal duration for each activity can be deduced. The  $T_{opt}$  parameter is thus easily definable for each activity. The different  $T_{max}$  parameters could also be defined thanks to *in vitro* observation. As shown in the modelling section, we need to set up the standard deviation to build a heterogeneous population of cells. These values do not have a biological representation concerning a cell model and fitting them is difficult. The first step of the modelling process consists in tuning these parameters to obtain a simulation with a constant evolution of the ratio of cells in each phase.

To help the biologists in this process a dedicated tool is built for cell cycle modelling. This tool offers to the biologists a means of visualization for the duration of each activity, taking into account its different parameters. The modeller can then adjust his parameter values in function of his knowledge of the temporal behaviour of the cell cycle being designed.

Figure 3 shows the kind of output offered by this tool. In this case the output is average duration of a dedicated activity function of its  $T_{opt}$  value and its rate of success. The modeller should use this feedback to adjust the different parameters until the different values suit his purpose.

The second step of the modelling process is to reproduce the experiment of mitosis unsynchronisation to verify if the population is not too heterogeneous. The expected be-

haviour of a population of cells in this kind of simulation can be extracted from *in vitro* culture. In other words, the modeller is able to define the number of cycles needed to observe a constant evolution in each phase. If the heterogeneity of the population is too high the unsynchronisation will occur too early in the simulation and further experiments would be biased by this behaviour.

The sequential aspect and the properties of the model suggest that this modelling protocol could be automatised. In further work we could try to express the different probabilities of transition between the different phases (i.e G1,S,G2,M) and try to find the best set of standard deviation parameters for an expected simulation result i.e a constant evolution of the ratio of cells in each phase.

The simplified environment will shortly be extended to a 2-D continuous environment and, finally, to a 3-D continuous environment. This will allow to reach the final aim of simulating the spatial organisation of multicellular tumour spheroids. As an intermediate step, all the 2-D monolayer classical experiments done *in vitro* will be reproduced *in silico*. This step will evaluate the response and the influence of the physical model by comparison between the results of the simulation with the proposed simplify environment and *in vitro* experiments.

Precisely, this 2-D prototype will be validated by evaluating the convergence of *in vitro* experiments and *in silico* simulation with specific scenarii. For example, we will use the following validation experiments: cell cycle synchronisation through a lack of environmental factors (arrest in G0); cell cycle synchronisation using a procedure known as double thymidine block (arrest at G1/S); application of a compound targeting the assembly of the microtubules (arrest at mitosis); etc.

## References

- Aguda, B. and Tang, Y. (1999). The kinetic origins of the restriction point in the mammalian cell cycle. *Cell proliferation*, 32(5):321–335.
- Calzone, L., Thieffry, D., Tyson, J., and Novak, B. (2007). Dynamical modeling of syncytial mitotic cycles in *Drosophila* embryos. *Molecular systems biology*, 3(1).
- Chavoya, A. and Duthen, Y. (2008). A cell pattern generation model based on an extended artificial regulatory network. *Biosystems*, 94(1-2):95–101.
- Chen, K., Calzone, L., Csikasz-Nagy, A., Cross, F., Novak, B., and Tyson, J. (2004). Integrative analysis of cell cycle control in budding yeast. *Molecular Biology of the Cell*, 15(8):3841.
- Cussat-Blanc, S., Pascalie, J., Luga, H., and Duthen, Y. (2010). Morphogen positioning by the means of a hydrodynamic engine. In *Artificial Life XII*. MIT Press, Cambridge, MA.
- Doursat, R. (2006). The growing canvas of biological development: Multiscale pattern generation on an expanding lattice of gene regulatory networks. *InterJournal: Complex Systems*, 1809.
- Drasdo, D. and Loeffler, M. (2001). Individual-based models to growth and folding in one-layered tissues: intestinal crypts and early development. *Nonlinear Analysis-Theory Methods and Applications*, 47(1):245–256.
- Elledge, S. (1996). Cell cycle checkpoints: preventing an identity crisis. *Science*, 274(5293):1664.
- Galle, J., Loeffler, M., and Drasdo, D. (2005). Modeling the effect of deregulated proliferation and apoptosis on the growth dynamics of epithelial cell populations *in vitro*. *Biophysical journal*, 88(1):62–75.
- Gerlee, P. and Anderson, A. (2007). An evolutionary hybrid cellular automaton model of solid tumour growth. *Journal of theoretical biology*, 246(4):583–603.
- Hoehme, S. and Drasdo, D. (2010). A cell-based simulation software for multi-cellular systems. *Bioinformatics*, 26(20):2641.
- Kauffman, S. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467.
- Loeffler, M. and Roeder, I. (2004). Conceptual models to understand tissue stem cell organization. *Current opinion in hematology*, 11(2):81.
- Meineke, F., Potten, C., and Loeffler, M. (2001). Cell migration and organization in the intestinal crypt using a lattice-free model. *Cell proliferation*, 34(4):253–266.
- Moreira, J. and Deutsch, A. (2002). Cellular automation models of tumor development: a critical review. *Advances in Complex Systems*, 5(2/3):247–268.
- Novak, B., Pataki, Z., Ciliberto, A., and Tyson, J. (2001). Mathematical model of the cell division cycle of fission yeast. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 11:277.
- Novak, B. and Tyson, J. (1993). Numerical analysis of a comprehensive model of M-phase control in *Xenopus* oocyte extracts and intact embryos. *Journal of Cell Science*, 106(4):1153–1168.
- Novak, B. and Tyson, J. (2004). A model for restriction point control of the mammalian cell cycle. *Journal of theoretical biology*, 230(4):563–579.
- Pascalie, J., Luga, H., Lobjois, V., Ducommun, B., and Duthen, Y. (2010). A checkpoint-orientated modeling for cell cycle simulation. In *Bionetics 10*.
- Patel, A. A., Gawlinski, E. T., Lemieux, S. K., and Gatenby, R. A. (2001). A cellular automaton model of early tumor growth and invasion: The effects of native tissue vascularity and increased anaerobic tumor metabolism. *Journal of Theoretical Biology*, 213(3):315 – 331.
- Pomeroy, J., Kim, S., and Ferrell Jr, J. (2005). Systems-level dissection of the cell-cycle oscillator: bypassing positive feedback produces damped oscillations. *Cell*, 122(4):565–578.
- Singhania, R., Sramkoski, R., Jacobberger, J., Tyson, J., and Beard, D. (2011). A Hybrid Model of Mammalian Cell Cycle Regulation. *PLoS Computational Biology*, 7(2):835–842.

# Robustness, Evolvability, and Accessibility in the Signal-Integration Space of Gene Regulatory Circuits

Joshua L. Payne<sup>1</sup> and Jason H. Moore<sup>1</sup>

<sup>1</sup>Dartmouth College, Hanover, NH 03755  
Joshua.Payne@Dartmouth.edu

## Abstract

Gene expression is commonly modulated by a set of regulating gene products, which bind to a gene's *cis*-regulatory region. This region encodes an input-output function, referred to as signal-integration logic, that maps a specific combination of regulatory signals (inputs) to a particular gene expression state (output). The space of all possible signal-integration functions (genotypes) is vast and highly redundant: for the same set of inputs, many functions yield the same expression output (phenotype). Here, we exhaustively characterize signal-integration space within a computational model of genetic regulation. Our goal is to understand how the inherent redundancy of signal-integration space affects the relationship between robustness and evolvability in regulatory circuits. Among a number of results, we show that robust phenotypes are (i) evolvable, (ii) easily identified by random mutation, and (iii) mutationally biased toward other robust phenotypes. We then explore the implications of these results for mutation-based evolution by conducting an ensemble of random walks between randomly chosen source and target phenotypes. We demonstrate that the time required to identify the target phenotype is independent of the properties of the source phenotype.

## Introduction

Living organisms exhibit two seemingly paradoxical properties: They are robust to genetic change, yet highly evolvable (Wagner, 2005). These properties appear contradictory because the former requires that genetic alterations leave the phenotype intact, while the latter requires these alterations to be used for the exploration of new phenotypes. Despite this apparent contradiction, several empirical analyses of living systems, particularly at the molecular scale, have revealed that robustness often facilitates evolvability (Bloom et al., 2006; Ferrada and Wagner, 2008; Isalan et al., 2008). In the cytochrome P450 BM3 protein, for example, increased protein stability — defined as the tendency of a protein to adopt its native structure in the face of mutation — increases the probability that mutants can exploit new substrates (Bloom et al., 2006).

To clarify the relationship between robustness and evolvability, several theoretical models have been proposed (*e.g.*, Newman and Engelhardt (1998); Wagner (2008a); Draghi

et al. (2010)). A common feature of these models is the concept of a genotype network (a.k.a. neutral network). In such a network, each node represents a genotype and edges connect genotypes that share the same phenotype and can be interconverted via single mutational events. In the case of RNA, for example, nodes represent DNA sequences and two nodes are connected if their corresponding sequences confer the same secondary structure, yet differ by a single nucleotide (Schuster et al., 1994). Large genotype networks thus correspond to robust phenotypes, where most mutations are neutral and therefore leave the phenotype unchanged. Phenotypic robustness confers evolvability because a population can diffuse neutrally throughout the genotype network (Huynen et al., 1996) and build up genetic diversity, which allows access to novel phenotypes through non-neutral point mutations into adjacent genotype networks (Wagner, 2008a).

Genotype networks have been used to explore the relationship between robustness and evolvability in a variety of biological systems, ranging from the molecular (Schuster et al., 1994; Cowperthwaite et al., 2008; Ferrada and Wagner, 2008; Wagner, 2008b) to the cellular level (Aldana et al., 2007; Ciliberti et al., 2007a,b; Mihaljev and Drossel, 2009). In the latter case, the phenotype of interest is typically a gene expression pattern and its corresponding genotype is a gene regulatory network, which consists of a structured set of gene products that activate and inhibit one another's expression. Gene expression is controlled by a gene's *cis*-regulatory region (Fig. 1A), which can be thought to perform a computation (Fig. 1B), using the regulating gene products as inputs. The regulatory program that encodes this computation is referred to as signal-integration logic.

Previous studies of the robustness and evolvability of gene regulatory networks have focused on the specific case where genetic perturbations alter network structure by adding or deleting regulatory interactions (Aldana et al., 2007; Ciliberti et al., 2007a,b; Mihaljev and Drossel, 2009). In this case, two gene regulatory networks are connected in the genotype network if they confer the same gene expression pattern, yet differ in a single regulatory interaction. The correspond-

ing genotype network is therefore a “network of networks” (Ciliberti et al., 2007b). These analyses have revealed several general properties of gene regulatory networks. First, robustness is an evolvable trait (Ciliberti et al., 2007b; Mihaljev and Drossel, 2009). Second, phenotypes are made up of vast genotype networks that span throughout the space of all possible genotypes (Ciliberti et al., 2007a; Mihaljev and Drossel, 2009); and third, highly robust phenotypes are often highly evolvable (Aldana et al., 2007; Ciliberti et al., 2007a).

While these studies have helped to elucidate the relationship between robustness and evolvability in gene regulatory networks, they are limited by their assumption that genetic perturbations primarily affect network structure. It is well known that the presence or absence of regulatory interactions is not the only determining factor of gene expression patterns (Setty et al., 2003; Mayo et al., 2006; Kaplan et al., 2008; Hunziker et al., 2010). By altering the arrangement of promoters and transcription factor binding sites (Fig. 1A, shaded boxes) in a gene’s *cis*-regulatory region, the signal-integration logic of gene regulation can be dramatically influenced. For example, by simply rearranging the location of transcription start sites in the promoter region of a reporter gene in the galactose network of *Escherichia Coli*, it is possible to generate 12 out of the 16 possible Boolean outputs (Hunziker et al., 2010). Thus, it is not only the structure of regulatory interactions that affects robustness and evolvability, but also the logic of signal-integration used in the *cis*-regulatory region of each gene. When genetic perturbations correspond to changes in the signal-integration logic, two gene regulatory networks are connected in the genotype network if they are topologically identical and confer the same gene expression pattern, yet differ in a single element of their signal-integration logic. The extent to which genetic perturbations in the signal-integration logic of gene regulatory networks affect robustness and evolvability remains largely unexplored. Further, the ease with which a phenotype is accessed by blind mutation, and how this relates to robustness and evolvability in the signal-integration logic of gene regulation, has not been addressed.

Here, we investigate the relationship between robustness and evolvability in the signal-integration logic of model gene regulatory circuits. These small circuits are ideal for this investigation because their genotype networks are exhaustively enumerable, which allows for a full characterization of the relationship between robustness and evolvability. To understand how robustness and evolvability influence mutation-based evolution, we conduct an ensemble of random walks between randomly chosen source and target phenotypes. We discuss the implications of our results and present directions for future work.

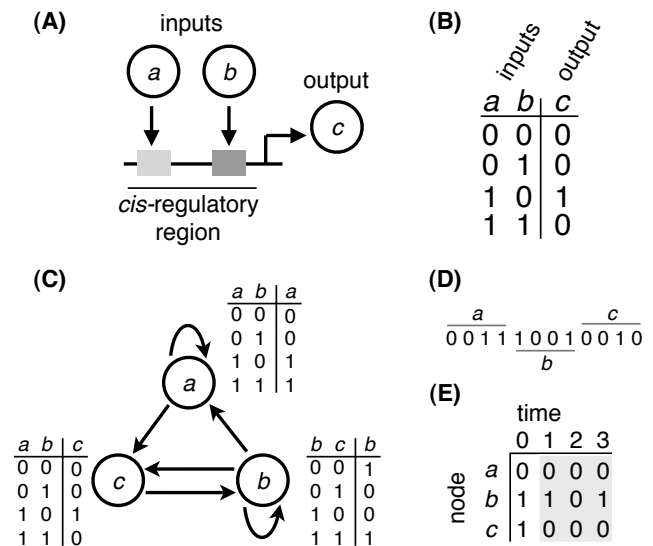


Figure 1: (A) Schematic of genetic regulation, where gene products  $a$  and  $b$  serve as regulatory inputs, attaching to their respective binding sites (gray shaded boxes) in the *cis*-regulatory region of gene  $c$  to influence its expression. The input-output function encoded in this regulatory region is called signal-integration logic and can be modeled as (B) a discrete function that explicitly maps all of the  $2^z$  input-output combinations of a  $z$ -input function. Here,  $z = 2$ . (C) All interactions between gene products  $a$ ,  $b$ , and  $c$  can be represented as a Random Boolean Circuit (RBC) with  $N = 3$  nodes. Gene product  $c$  possesses the same regulatory inputs and signal-integration logic as in (A) to clearly depict how the RBC abstraction captures genetic regulation. (D) The signal-integration logic of every node in the RBC can be simultaneously represented with a single rule vector by concatenating the rightmost columns of each node’s look-up table. (E) The dynamics of the RBC begin with an initial state (e.g.,  $\langle 011 \rangle$ ) and eventually settle into an attractor (gray shaded region).

## Methods

### Random Boolean Circuits

We use Random Boolean Circuits (RBCs) to model genetic regulation (Kauffman, 1969). RBCs are composed of nodes and directed edges (Fig. 1C). Nodes represent gene products and edges represent regulatory interactions. Two nodes  $a$  and  $c$  are connected by a directed edge  $a \rightarrow c$  if the expression of gene  $c$  is regulated by gene product  $a$ . Node states are binary, reflecting the presence (1) or absence (0) of a gene product, and dynamic, such that the state of a node at time  $t + 1$  is dependent upon the states of its regulating nodes at time  $t$ . This dependence is captured by a look-up table associated with each node, which explicitly maps all possible combinations of regulatory input states to an output expression state. This look-up table is analogous to the

signal-integration logic encoded in *cis*-regulatory regions. The signal-integration logic of all of the nodes in the network can be simultaneously represented using a single *rule vector* (Fig. 1D).

The dynamics of RBCs occur in discrete time with synchronous updating of node states (Fig. 1E). The dynamics begin at a pre-specified initial state, which can be thought to represent regulatory factors upstream of the circuit (Ciliberti et al., 2007a; Martin and Wagner, 2009). The dynamics then unfold according to the circuit's structure and signal-integration logic. Since the system is both finite and deterministic, its dynamics eventually settle into an attractor (Kauffman, 1969), which represents the gene expression pattern, and is referred to as the phenotype. We refer to the combination of circuit structure, rule vector, and initial state as an *instance* of a RBC.

While simple, the Boolean abstraction has proven capable of precisely replicating specific properties of genetic regulation in natural systems. For example, variants of the model have emulated the expression patterns of the fruit fly *Drosophila melanogaster* (Albert and Othmer, 2003), the plant *Arabidopsis thaliana* (Espinosa-Soto et al., 2004), and the yeast *Saccharomyces pombe* (Davidich and Bornholdt, 2008). Due to their accuracy in capturing the dynamics of genetic regulation, and because the signal-integration logic of each gene is explicitly represented, RBCs are ideal synthetic systems for investigating the relationship between robustness and evolvability when genetic perturbations correspond to changes in signal-integration logic.

## Dynamical Regimes of RBCs

An important feature of RBCs is that they exhibit three dynamical regimes: ordered, critical, and chaotic (Kauffman, 1969). In the ordered regime, gene expression patterns are relatively insensitive to perturbations, while in the chaotic regime they are highly sensitive. The critical regime delineates these two extremes. For randomly constructed circuits, the transitions between regimes are controlled by two parameters: the average in-degree  $z$  and the probability  $\rho$  of gene expression (*i.e.*, the probability of observing a 1 in the rule vector). Letting  $S = 2\rho(1 - \rho)z$ , the RBC lies in the ordered regime when  $S < 1$ , the critical regime when  $S = 1$ , and the chaotic regime when  $S > 1$ . When there is an equal probability of observing a 0 or a 1 in the rule vector ( $\rho = 0.5$ ) the dynamical regime is determined solely by the average in-degree, with  $z < 2$  yielding the ordered regime,  $z = 2$  the critical regime, and  $z > 2$  the chaotic regime. In this study,  $\rho = 0.5$ .

## Genotype Networks

We refer to the signal-integration logic of a RBC, as represented by its rule vector (Fig. 1D), as the genotype. There are a total of  $2^L$  unique genotypes for a given combination of circuit structure and initial state, where  $L = N2^z$ . We refer

to this set of genotypes as the genotype space, or equivalently, as the signal-integration space. For the RBCs considered here, the size of the genotype space ranges from  $2^6$  for the ordered regime to  $2^{24}$  for the chaotic regime.

These genotypes map to a significantly smaller set of phenotypes. This high level of redundancy is a general feature of RBCs, and can be formalized using a genotype network, in which rule vectors are represented as nodes, and edges connect rule vectors that differ by a single bit, yet yield the same gene expression pattern (*i.e.*, phenotype). Thus, we define a neutral point mutation as a single change to an element of the genotype that does not lead to a change in phenotype. Such a mutation is analogous to a change in the position of a transcription factor binding site in the *cis*-regulatory region that leaves the gene expression pattern unchanged. Genotype networks are measured using an exhaustive breadth-first search, thus discovering all genotypes that yield the same phenotype and are accessible via neutral point mutations, starting from the original genotype of the RBC instance.

The quantity  $v_{ij}$  captures the number of unique non-neutral point mutations to genotypes in the genotype network of phenotype  $i$  that lead to genotypes in the genotype network of phenotype  $j$ . We call phenotypes  $i$  and  $j$  adjacent if  $v_{ij} > 0$ . By enumerating all of the phenotypes that are adjacent to phenotype  $i$ , and their corresponding genotype networks, we capture the mutational biases between adjacent phenotypes.

## Robustness, Evolvability, and Accessibility

Several definitions of robustness and evolvability have been proposed, at both the genotypic and phenotypic scales (Alldana et al., 2007; Wagner, 2008b; Mihaljev and Drossel, 2009; Draghi et al., 2010). Here, we focus on these properties at the level of the phenotype. We define robustness  $R_i$  as the proportion of signal-integration space occupied by the genotype network of phenotype  $i$ . This metric is independent of rule vector length  $L$ , and captures the fraction of all genotypes that yield the same phenotype and can be accessed via neutral point mutations.

We define evolvability using two metrics. The first  $E_{1,i}$  is simply the number of phenotypes that can be accessed through non-neutral point mutations from the genotype network of phenotype  $i$  (Wagner, 2008b). The second  $E_{2,i}$  captures the mutational biases that exist between the genotype networks of adjacent phenotypes (Cowperthwaite et al., 2008). Letting

$$f_{ij} = \frac{v_{ij}}{\sum_{k \neq i} v_{ik}} \quad (1)$$

denote the fraction of non-neutral point mutations to genotypes of phenotype  $i$  that result in genotypes of phenotype  $j$ , we define the evolvability  $E_{2,i}$  of phenotype  $i$  as

$$E_{2,i} = 1 - \sum_j f_{ij}^2. \quad (2)$$



Since  $\sum_j f_{ij}^2$  captures the probability that two randomly chosen non-neutral point mutations to genotypes of phenotype  $i$  result in genotypes with identical phenotypes, its complement  $E_{2,i}$  captures the probability that these same mutations result in genotypes with distinct phenotypes. This metric takes on high values when a phenotype is adjacent to many other phenotypes and its non-neutral point mutations are uniformly divided amongst these phenotypes. The metric takes on low values when a phenotype is adjacent to only a few other phenotypes and its non-neutral point mutations are biased toward a subset of these phenotypes.

In addition to measuring evolvability, which captures the uniformity of non-neutral mutations from phenotype  $i$  into adjacent phenotypes, we also consider accessibility

$$A_i = \sum_j f_{ji}, \quad (3)$$

which captures the propensity to mutate into phenotype  $i$  (Cowperthwaite et al., 2008). This metric takes on high values if the phenotypes adjacent to phenotype  $i$  are mutationally biased toward  $i$  and low values otherwise.

Lastly, we measure the robustness of all phenotypes that are adjacent to phenotype  $i$ , in proportion to the probability that these phenotypes are encountered through a randomly chosen, non-neutral point mutation from phenotype  $i$  (Cowperthwaite et al., 2008). We refer to this quantity as adjacent robustness,

$$B_i = \sum_j f_{ij} \times R_j. \quad (4)$$

This metric takes on high values when a phenotype is mutationally biased toward robust phenotypes and low values otherwise.

## Simulation Details and Data Analysis

For all RBC instances, the rule vector and initial state are generated at random with  $\rho = 0.5$ . The circuit structure is also generated at random, but subject to the constraint that each node has exactly  $z$  inputs. Self-loops are permitted, mimicking autoregulation. We separately consider RBCs in the ordered, critical, and chaotic regimes by setting  $z = 1, 2, 3$ , respectively. The initial state and circuit structure are held fixed for each RBC instance. To ensure that all of the genotype networks considered in this study are amenable to exhaustive enumeration, we restrict our attention to RBCs with  $N = 3$  nodes. While small, sensitivity analysis (Derrida and Pomeau, 1986) confirms that these RBCs exhibit the same dynamical regimes as larger networks, albeit with shorter attractors. To assess the strength and significance of the trends in our data, we employ Pearson's correlation coefficient.

## Results

### Characteristics of Genotype Networks

To characterize the genotype networks of signal-integration space in RBCs, we randomly generate 2500 RBC instances for each dynamical regime and exhaustively characterize the genotype networks of their corresponding phenotypes, and the genotype networks of all adjacent phenotypes.

The range of phenotypic robustness  $R$  varies with dynamical regime, with ordered RBCs spanning the smallest range ( $3.12 \times 10^{-2} \leq R \leq 1.25 \times 10^{-1}$ ), critical RBCs spanning an intermediate range ( $4.88 \times 10^{-4} \leq R \leq 1.25 \times 10^{-1}$ ), and chaotic RBCs spanning the largest range ( $1.19 \times 10^{-7} \leq R \leq 1.25 \times 10^{-1}$ ). The maximum value of phenotypic robustness is independent of dynamical regime, and corresponds to fixed-point attractors. Since these attractors comprise a single state, only  $N$  bits of the rule vector are accessed during the RBC's dynamics, leaving  $L - N$  bits unused. Thus, the corresponding genotype network is of size  $2^{L-N}$ , with phenotypic robustness  $R_{\max} = 2^{-N} = 1.25 \times 10^{-1}$ . The average phenotypic robustness decreases from the ordered ( $R = 9.44 \times 10^{-2}$ ) to the critical ( $R = 4.12 \times 10^{-2}$ ) to the chaotic ( $R = 3.02 \times 10^{-2}$ ) regime.

Evolvability  $E_1$  and phenotypic robustness  $R$  are positively correlated (Fig. 2A), and the strength of correlation increases from the ordered ( $r = 0.75, p \ll 0.01$ ) to the critical ( $r = 0.90, p \ll 0.01$ ) to the chaotic ( $r = 0.98, p \ll 0.01$ ) regime. This indicates that, in this system, no trade-off exists between robustness and the number of phenotypes accessible via non-neutral point mutations; the more robust the phenotype, the higher its evolvability. Average evolvability  $E_1$  increases faster than linearly with increasing  $z$ , indicating a rapid increase in the number of adjacent phenotypes as the dynamical regime shifts from ordered to chaotic (Fig. 2A, inset).

When mutational biases between adjacent phenotypes are taken into account using  $E_2$ , a slightly different relationship is observed between evolvability and phenotypic robustness (Fig. 2B). RBCs in the ordered regime exhibit a weak and insignificant correlation between  $E_2$  and  $R$  ( $r = 0.02, p = 0.41$ ). In contrast, RBCs in the critical and chaotic regimes exhibit weak, but significant correlations, with the strength of correlation increasing from the critical ( $r = 0.10, p \ll 0.01$ ) to the chaotic regime ( $r = 0.42, p \ll 0.01$ ). The average value of  $E_2$  increases approximately linearly as  $z$  increases (Fig. 2B, inset). Thus, the average probability that two randomly chosen, non-neutral point mutations lead to distinct phenotypes is only  $\approx 15\%$  higher in chaotic RBCs than in ordered RBCs, despite the four order-of-magnitude difference in the absolute number of adjacent phenotypes (Fig. 2A, inset).

Accessibility  $A$  and phenotypic robustness  $R$  are positively correlated (Fig. 2C), with the strength of correlation again increasing from the ordered ( $r = 0.88, p \ll 0.01$ )

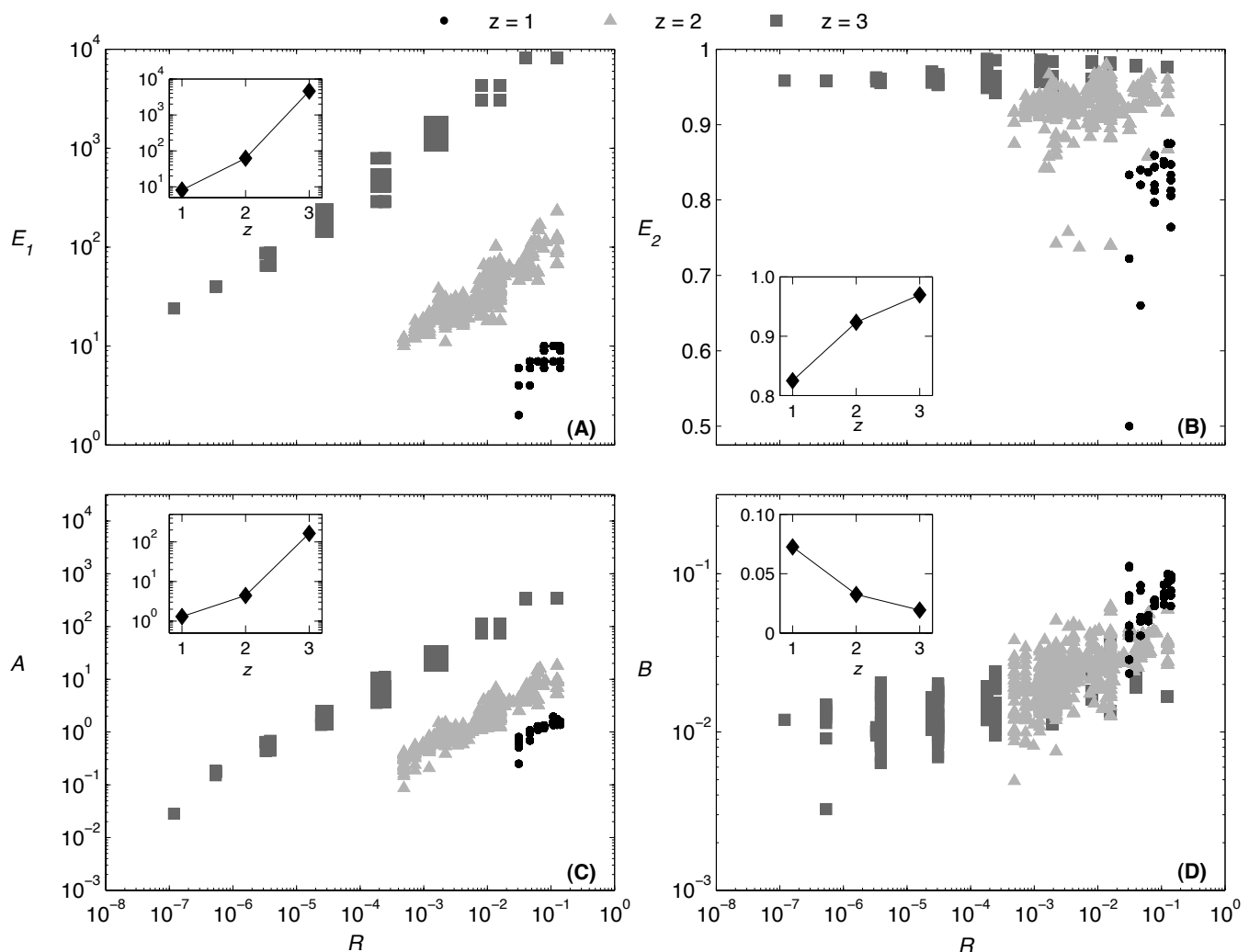


Figure 2: (A,B) Evolvability, (C) accessibility, and (D) adjacent robustness as a function of phenotypic robustness  $R$  for each of the three dynamical regimes: ordered ( $z = 1$ ), critical ( $z = 2$ ), and chaotic ( $z = 3$ ). Each data point represents one of 2500 RBC instances for each dynamical regime. The insets depict the corresponding averages, as a function of  $z$ . Lines are provided as a guide for the eye.

to the critical ( $r = 0.94, p \ll 0.01$ ) to the chaotic ( $r = 0.98, p \ll 0.01$ ) regimes. This implies that, for all three dynamical regimes, random point mutations are more likely to lead to robust phenotypes than to non-robust phenotypes. Average accessibility increases faster than linearly as  $z$  increases (Fig. 2C, inset), indicating a rapid increase in the relative ease with which phenotypes are found by random mutation as the dynamical regime shifts from ordered to chaotic.

Adjacent robustness  $B$  and phenotypic robustness  $R$  are positively correlated, with the strength of correlation decreasing from the ordered ( $r = 0.81, p \ll 0.01$ ) to the critical ( $r = 0.66, p \ll 0.01$ ) to the chaotic regimes ( $r = 0.35, p \ll 0.01$ ). This implies that non-neutral point mutations to genotypes within robust phenotypes often lead to

other robust phenotypes, but that the strength of this tendency weakens as RBCs approach the chaotic regime. The average adjacent robustness  $B$  decreases approximately linearly as  $z$  increases (Fig. 2D, inset), indicating that the expected robustness of a phenotype encountered via non-neutral point mutation decreases as the dynamical regime shifts from ordered to chaotic.

Taken together, these results suggest that a series of random point mutations will tend toward phenotypes of increased robustness (Fig. 2D) and correspondingly increased evolvability (Fig. 2A,B). Further, the ease with which such a blind evolutionary process identifies an arbitrary phenotype should increase with that phenotype's robustness (Fig. 2C) and as the dynamical regime shifts from ordered to critical to chaotic (Fig. 2C, inset).

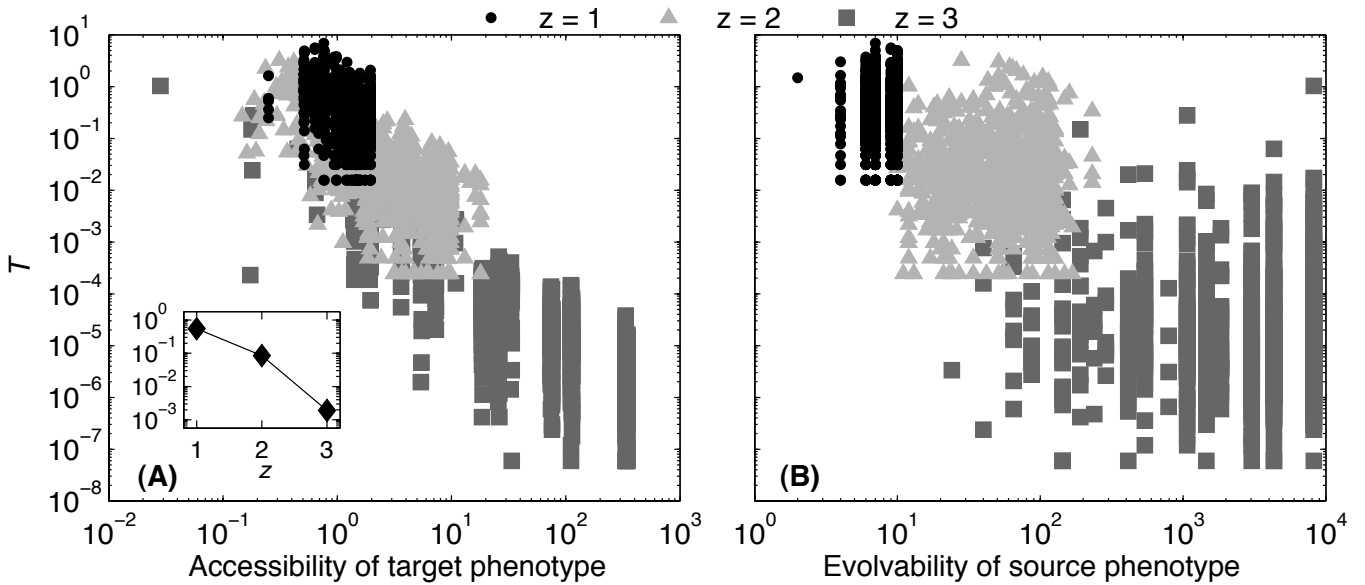


Figure 3: Waiting time of a random walk  $T = S/2^L$  as a function of (A) the target phenotype's accessibility  $A$  and (B) the source phenotype's evolvability  $E_1$ , for each of the three dynamical regimes: ordered ( $z = 1$ ), critical ( $z = 2$ ), and chaotic ( $z = 3$ ). The inset in (A) depicts the average waiting time  $T$  as a function of  $z$ . Lines are provided as a guide for the eye.

### Random Walks Through Signal-Integration Space

To investigate how robustness, evolvability, and accessibility influence blind, mutation-based evolution, we conduct an ensemble of random walks. For each dynamical regime, we randomly generate 1000 RBC instances and identify the phenotype of each instance as a source phenotype. For each instance, we then sample the genotype space at random until we discover a genotype that yields a different phenotype from the source phenotype, and we identify this as the target phenotype. For each pair of source and target phenotypes, we then perform a random walk, starting from the instance's genotype and ending when the random walk reaches any genotype in the target phenotype. Each step in the random walk corresponds to a single point mutation to the genotype. We record the number of steps  $S$  required to reach the target phenotype, which we normalize by the size of the signal-integration space  $2^L$ , and refer to as the waiting time  $T = S/2^L$ .

Waiting time  $T$  decreases faster than linearly as  $z$  increases (Fig. 3A, inset). For all three dynamical regimes, waiting time is strongly negatively correlated with the accessibility  $A$  of the target phenotype (Fig. 3A), and the strength of correlation increases from the ordered ( $r = -0.41, p \ll 0.01$ ) to the critical ( $r = -0.67, p \ll 0.01$ ) to the chaotic ( $r = -0.82, p \ll 0.01$ ) regime. In contrast, the correlation between waiting time  $T$  and the evolvability  $E_1$  of the source phenotype is weak and insignificant ( $z = 1 : r = -0.03, p = 0.38$ ;  $z = 2 : r = 0.01, p = 0.82$ ;  $z = 3 : r = -0.02, p = 0.56$ ) (Fig. 3B). Similarly weak and insignificant correlations were observed between wait-

ing time  $T$  and other characteristics of the source phenotype, such as  $E_2$ ,  $A$ , and  $B$ . These results indicate that the time required for a blind evolutionary search to identify a target phenotype is independent of the phenotypic properties of the starting point and solely dependent upon the phenotypic properties of the target.

### Discussion

This study has provided the first characterization of genotype networks in the signal-integration space of Random Boolean Circuits (RBCs), highlighting the relationship between robustness and the evolvability and accessibility of phenotypes. We found a positive correlation between robustness and evolvability, as measured by either the absolute number of adjacent phenotypes  $E_1$  (Fig. 2A) or by the probability that two non-neutral point mutations lead to distinct phenotypes  $E_2$  (Fig. 2B). Our results corroborate the observation made in previous studies that gene regulatory networks can simultaneously exhibit robustness and evolvability (Aldana et al., 2007; Ciliberti et al., 2007a,b). Further, our analyses extend these previous studies by providing an explicit description of this relationship and by considering genetic perturbations that alter the signal-integration logic encoded in *cis*-regulatory regions, instead of genetic perturbations that alter circuit structure.

We also found a positive correlation between robustness and accessibility (Fig. 2C), a measure that captures the relative ease with which a phenotype can be identified by mutation-based evolution. This result supports the intuitive notion that phenotypes comprising many genotypes are eas-

ier for evolution to identify than those comprising few genotypes. In addition, robust phenotypes are mutationally biased toward other robust phenotypes (Fig. 2D), indicating that the robustness of phenotypes encountered by blind mutation-based evolution should, on average, tend to increase.

To understand how phenotypic robustness, evolvability, and accessibility in signal-integration space influence mutation-based evolution, we considered an ensemble of random walks between pairs of source and target phenotypes. We found that the number of random mutations required to reach the target phenotype was entirely dependent upon its accessibility (Fig. 3A) and independent of any properties of the source phenotype (Fig. 3B). This suggests that a random walk through signal-integration space quickly loses any memory of its starting location. Consequently, extant evolvability metrics cannot be expected to predict the duration of a random walk between phenotypes.

The majority of our results are consistent with those made in RNA systems (Cowperthwaite et al., 2008; Wagner, 2008b). However, there is one difference worth emphasizing: the correlation between robustness and evolvability  $E_2$  is negative in RNA (Cowperthwaite et al., 2008). Since the relationship between robustness and adjacent robustness  $B$  is positive in RNA systems, Cowperthwaite et al. (2008) concluded that robust phenotypes act as “evolutionary traps.” That is, random mutation will tend toward phenotypes of higher robustness, which in turn are less evolvable, and therefore stagnate evolutionary search. Since we observed a positive correlation between (i) robustness and evolvability  $E_2$  and (ii) robustness and adjacent robustness  $B$ , we conclude that robust phenotypes in the signal-integration space of RBCs are not evolutionary traps, but instead facilitate the discovery of novel phenotypes. Such contrast between model systems highlights the fact that the relationships between robustness, evolvability, and accessibility are system dependent.

Evolvability increased monotonically as  $z$  increased (Fig. 2A,B, insets) and the maximum achievable robustness was independent of  $z$  ( $R_{\max} = 2^{-N}$ ). Taken together, these results indicate that robustness and evolvability can be simultaneously maximized in chaotic RBCs. This result contrasts with previous analysis (Aldana et al., 2007), which found robustness and evolvability to be simultaneously maximized in critical RBCs. This discrepancy can be understood by considering the two primary differences between the analyses. First, Aldana et al. (2007) focused on genetic perturbations that altered circuit structure (and consequently, in some cases, signal-integration logic) while we focused solely on genetic perturbations that altered signal-integration logic. Second, and of greater importance, the measures of robustness and evolvability considered by Aldana et al. (2007) were not based on genotype networks. Instead, robustness was defined as the ability of a single mu-

tated genotype to maintain the phenotypic landscape (*i.e.*, the set of all phenotypes observed across all possible initial states), and evolvability was defined as the capacity of the mutated genotype to expand the phenotypic landscape (*i.e.*, add new phenotypes to the set of existing phenotypes). Thus, Aldana et al. (2007) focused on robustness and evolvability at the level of the genotype rather than the phenotype (Wagner, 2008b). While these definitions are reasonable and insightful, our departure from their use precludes any direct comparison between the two studies. That said, our observation that chaotic RBCs simultaneously optimize robustness and evolvability must be interpreted with caution. For all dynamical regimes, robustness is maximal for fixed point attractors, and these occur with decreasing frequency as  $z$  increases. Thus, while it is only possible to simultaneously observe maximal robustness and maximal evolvability in chaotic RBCs, this case represents the exception rather than the rule.

Future work will seek to understand how evolution navigates signal-integration space. Is it possible for mutation and selection to identify the high-robustness, high-evolvability phenotypes of chaotic RBCs? If so, can they out-compete critical and ordered RBCs in static (Oikonomou and Cluzel, 2006) or dynamic (Greenbury et al., 2010) environments? How are these evolutionary outcomes affected by mutation rate (Wilke et al., 2001) or recombination (Martin and Wagner, 2009)? Future research will also focus on larger systems, moving from an analysis of circuits to entire networks. To accomplish this, Monte Carlo sampling methods will be required (Jörg et al., 2008), as the increased size of the signal-integration space will prohibit the exhaustive enumeration of genotype networks. In addition, future work will seek to understand both the influence of canalizing functions (Kauffman et al., 2004) and the probability of gene expression  $\rho$  on the size and structure of genotype networks. These directions, among others, will lead to a more thorough understanding of how the genetic flexibility of *cis*-regulatory regions influence evolutionary processes.

## Acknowledgements

The authors would like to thank Davnah Urbach and Dov Pechenick for their insightful comments, which greatly improved the clarity and presentation of this manuscript. This work was partially supported by NIH grants R01-LM009012, R01-LM010098, and R01-AI59694. J.L.P. was supported by NIH grant K25-CA134286.

## References

- Albert, R. and Othmer, H. G. (2003). The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*. *Journal of Theoretical Biology*, 223:1–18.
- Aldana, M., Balleza, E., Kauffman, S., and Resendiz, O. (2007). Robustness and evolvability in genetic regulatory networks. *Journal of Theoretical Biology*, 245:433–448.

- Bloom, J. D., Labthavikul, S. T., Otey, C. R., and Arnold, F. H. (2006). Protein stability promotes evolvability. *PNAS*, 103:5869–5874.
- Ciliberti, S., Martin, O. C., and Wagner, A. (2007a). Innovation and robustness in complex regulatory gene networks. *PNAS*, 104:13591–13596.
- Ciliberti, S., Martin, O. C., and Wagner, A. (2007b). Robustness can evolve gradually in complex regulatory gene networks with varying topology. *PLoS Computational Biology*, 3:e15.
- Cowperthwaite, M. C., Economo, E. P., Harcombe, W. R., Miller, E. L., and Meyers, L. A. (2008). The ascent of the abundant: how mutational networks constrain evolution. *PLoS Computational Biology*, 4:e10000110.
- Davidich, M. I. and Bornholdt, S. (2008). Boolean network model predicts cell cycle sequence of fission yeast. *PLoS ONE*, 3:e1672.
- Derrida, B. and Pomeau, Y. (1986). Random networks of automata: a simple annealed approximation. *Europhysics Letters*, 1:45–49.
- Draghi, J. A., Parsons, T. L., Wagner, G. P., and Plotkin, J. B. (2010). Mutational robustness can facilitate adaptation. *Nature*, 463:353–355.
- Espinosa-Soto, C., Padilla-Longoria, P., and Alvarez-Buylla, E. R. (2004). A gene regulatory network model for cell-fate determination during *Arabidopsis thaliana* flower development that is robust and recovers experimental gene expression profiles. *The Plant Cell*, 16:2923–2939.
- Ferrada, E. and Wagner, A. (2008). Protein robustness promotes evolutionary innovations on large evolutionary time-scales. *Proceedings of the Royal Society London B*, 275:1595–1602.
- Greenbury, S. F., Johnston, I. G., Smith, M. A., Doye, J. P. K., and Louis, A. A. (2010). The effect of scale-free topology on the robustness and evolvability of genetic regulatory networks. *Journal of Theoretical Biology*, 267:48–61.
- Hunziker, A., Tuboly, C., Horváth, P., Krishna, S., and Semsey, S. (2010). Genetic flexibility of regulatory networks. *PNAS*, 107:12998–13003.
- Huynen, M. A., Stadler, P. F., and Fontana, W. (1996). Smoothness within ruggedness: The role of neutrality in adaptation. *PNAS*, 93:397–401.
- Isalan, M., Lemerle, C., Michalodimitrakis, K., Horn, C., Beltrao, P., Raineri, E., Garriga-Canut, M., and Serrano, L. (2008). Evolvability and hierarchy in rewired bacterial gene networks. *Nature*, 452:840–846.
- Jörg, T., Martin, O. C., and Wagner, A. (2008). Neutral network sizes of biological RNA molecules can be computed and are not atypically small. *BMC Bioinformatics*, 9:464.
- Kaplan, S., Bren, A., Zaslaver, A., Dekel, E., and Alon, U. (2008). Diverse two-dimensional input functions control bacterial sugar genes. *Molecular Cell*, 29:786–792.
- Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22:437–467.
- Kauffman, S. A., Peterson, C., Samuelsson, B., and Troein, C. (2004). Genetic networks with canalizing Boolean rules are always stable. *Proceedings of the National Academy of Sciences*, 101:17102–17107.
- Martin, O. C. and Wagner, A. (2009). Effects of recombination on complex regulatory circuits. *Genetics*, 138:673–684.
- Mayo, A. E., Setty, Y., Shavit, S., Zaslaver, A., and Alon, U. (2006). Plasticity of the *cis*-regulatory input function of a gene. *PLoS Biology*, 4:e45.
- Mihaljev, T. and Drossel, B. (2009). Evolution of a population of random Boolean networks. *European Physics Journal B*, 67:259–267.
- Newman, M. and Engelhardt, R. (1998). Effects of selective neutrality on the evolution of molecular species. *Proceedings of the Royal Society London B*, 265:1333–1338.
- Oikonomou, P. and Cluzel, P. (2006). Effects of topology on network evolution. *Nature Physics*, 2:532–536.
- Schuster, P., Fontana, W., Stadler, P. F., and Hofacker, I. L. (1994). From sequences to shapes and back: a case study in RNA secondary structures. *Proceedings of the Royal Society London B*, 255:279–284.
- Setty, Y., Mayo, A., Surette, M., and Alon, U. (2003). Detailed map of a *cis*-regulatory input function. *PNAS*, 100:7702–7707.
- Wagner, A. (2005). *Robustness and Evolvability in Living Systems*. Princeton University Press.
- Wagner, A. (2008a). Neutralism and selectionism: a network-based reconciliation. *Nature Reviews Genetics*, 9:965–974.
- Wagner, A. (2008b). Robustness and evolvability: a paradox resolved. *Proceedings of the Royal Society London B*, 275:91–100.
- Wilke, C. O., Wang, J. L., Ofria, C., Lenski, R. E., and Adami, C. (2001). Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, 412:331–333.

# Formalizing Institutions as Executable Petri Nets for Distributed Robotic Systems

José N. Pereira<sup>1,2</sup>, Porfírio Silva<sup>2</sup>, Pedro U. Lima<sup>2</sup> and Alcherio Martinoli<sup>1</sup>

<sup>1</sup> Distributed Intelligent Systems and Algorithms Laboratory, EPFL, Lausanne, Switzerland

<sup>2</sup> Intelligent Robots and Systems Group, Institute for Systems and Robotics, Lisbon, Portugal

<sup>1</sup> {jose.pereira | alcherio.martinoli}@epfl.ch, <sup>2</sup> {porfriosilva | pal}@isr.ist.utl.pt

## Abstract

Institutional Robotics is a new approach to the coordination of distributed robotic systems, drawing inspiration from social sciences. It aims to provide a comprehensive strategy for specifying social interactions among robots in the form of institutions. In this paper, we present a formalism for institutions in the Institutional Robotics model. We apply this formalism to two case studies. The first is concerned with a swarm of simple robots which has to maintain wireless connectivity. The second focuses on role allocation in a robotic team aimed at improving coordination and performance in a transportation task.

## Introduction

Multi-robot systems are nowadays an important area of research within the broader field of robotics. Using multiple robots might enhance the overall system performance not only because of a faster task execution speed but also in terms of robustness to failures and flexibility in allocation of subtasks. It is also clear that a team of robots is capable of completing some tasks that are impossible for single robots, for instance, because of their physical limitations. However, in order to leverage these potential benefits, it is not enough to add robots to the team. Cooperative behavior has to be present, and therefore interactions among robots must be coordinated in some way.

Institutional Robotics (IR) (Silva and Lima (2007)) is a new approach to the coordination of distributed robotic systems, drawing some inspiration from social sciences, namely from Institutional Economics' concepts. It combines the notions of institution, coordination artifact, and environment, aiming to provide a comprehensive strategy for specifying social interactions (e.g., norms, roles, hierarchies) among robots. In order to do so, robots are situated not only in a physical but also in an institutional environment, where their interactions are guided by institutions. Through cooperative decision-making, these institutions can be modified by the robots, providing adaptation to a changing scenario. Coordination is achieved by this regulation of social interactions since the robots know not only how to be-

have in a given scenario but also what to expect from other robots and the environment.

One of the goals of our research is to formalize the concepts of IR from a computer science perspective, so as to create an ontology of the entities that will be part of the IR model, and to describe ways of interconnecting them (such as graphs and tuples describing the entities associated to each node), as well as algorithms to manage a robotic collective based on social science principles.

In this work, we focus on formalizing the central concept of IR - institutions. Institutions are coordination artifacts specifying social interactions of different types and encapsulating relevant behavioral rules (possibly designed based on problem-domain knowledge) that, once adopted, avoid the need for the behavior to be re-learned or re-acquired. Our goal is to formalize them using an abstract representation, that will allow us to design these coordination artifacts and execute them in robots (both in reality and simulation), so as to obtain behaviors capturing the social interactions of interest. In order to accomplish this objective we propose to use Petri Nets as an abstract representation for institutions. Our method will produce, from a set of institutions, a robot controller able to execute a desired task.

We apply this formalism to two case studies. The first is concerned with a swarm of simple robots which has to maintain wireless connectivity. The second focuses on role allocation in a robotic team aimed at improving coordination and performance in a transportation task.

In Section 2 we discuss related work and motivation for our formalization. This formalization is presented in Section 3 culminating with the definition of a controller based on our institutional approach. In Section 4 and 5 we apply this formalism to two different case studies.

## Related Work

Institutional economics is a fundamentally different approach from neo-classical theory, the current trend of economics and inspiration for market-based systems of task allocation in distributed robotics (Dias et al. (2006)).

In Hodgson (2000), the author refines a description of in-

stitutional economics outlining the following main features: institutions are the key element of any economy; the economy is an open and evolving system; and the notion of individuals as utility-maximizing agents is inadequate. The institutional approach is characterized also by the rejection of unbounded rationality. Agents are affected by the institutional environment they live in, but in no way does that environment fully determine their behavior. Every agent has individual goals and motivations that it wants to fulfill. Institutions are developed by these very same agents.

In Crawford and Ostrom (1995) and Ostrom (2005), the authors propose a formal “grammar” of institutions according to the New Institutional Economics (NIE) approach. NIE is a compromise between the institutional and neo-classical theories of economics. Therein, the authors study what are the elements that compose institutional statements. While at this point most of these elements are not ready to be applied to multi-robot systems, deontic operators are fundamental in our IR version in order to specify how institutions relate to one another.

IR (Silva and Lima (2007)) aims to provide a comprehensive strategy for specifying social interactions among robots, by combining the notions of institution, coordination artifact, and environment. According to the IR approach:

1. the coordination strategy is supported by a network of institutions;
2. institutions are coordination artifacts of different types (e.g., norms, roles, hierarchies);
3. robots are able to modify both their physical and their institutional environment;
4. robots need a high degree of autonomy, pursuing goals based on their “struggle for survival”.

From an institutional perspective, institutions are taken as the main tool of any sophisticated society, and individuals are both constructive within and constructed through institutional environments. In a first attempt at formalizing institutions in the IR model, Silva et al. (2008) define them as “cumulative sets of persistent artificial modifications made to the environment or to the internal mechanisms of a subset of agents, thought to be functional to the collective order”.

This definition is too abstract to be applied “as is” to distributed robotics experiments. Thus, we go back to the idea of institutions as coordination artifacts (Tummolini and Castelfranchi (2006)). Coordination artifacts (Omicini et al. (2004); Ricci et al. (2005)) are infrastructure abstractions in multi-agent systems meant to improve the synthesis and analysis of coordination activities. The main properties that describe coordination artifacts are: *specialization*, *encapsulation*, and *inspectability*. Specialization refers to the fact that coordination artifacts are specialized in automating coordination activities and can be represented with concurrency frameworks such as Petri Nets or process algebras.

Coordination artifacts encapsulate a coordination service, allowing the agents to abstract how it is implemented. Encapsulation is the key to achieve reuse of coordination. Inspectability refers to the property that an artifact should support some procedure to allow engineers or agents responsible for the system to check for errors in its specification.

Omicini et al. argue that coordination artifacts are exterior to the agents using them and perceived as individual entities, but can actually be distributed on several nodes of a multi-agent system. We propose that, when taking institutions as coordination artifacts, they can be part of the agent controller, working as norms or procedures the agent has to follow. Even with this assumption, we can still think of institutions being distributed in our multi-robot system, if we consider their representation to be replicated in each agent.

## Petri Nets and Institutions

Starting from the concept of institutions as coordination artifacts we model them using a formal representation, leading to a standard design and execution platform (in real robots, realistic simulations, and multi-agent systems). Considering the three main properties of coordination artifacts mentioned above, we propose to use Petri Nets as formal framework.

Our choice of Petri Nets is based mostly on the ability of this formalism to deal with distributed systems. State information is distributed among a set of places that capture key conditions that govern the operation of the system. Moreover, Petri Nets not only are able to deal with distributed systems but are also a suitable computational model for effective and efficient interaction management, a key aspect of coordination artifacts. Finally, Petri Nets also have a larger representational power than Finite State Automata (FSA), being able to represent, with finite structure, languages that are not representable by FSA (Cassandras and Lafortune (2008)).

The Petri Net Plans (PNP) language is a tool specifically directed to the design and execution of robotic plans using Petri Nets (Ziparo et al. (2010)). Therein, properties of safety and liveness of PNs are used to ensure that execution of robotic tasks in robots follows the designed plan. However, these properties can also be verified on simpler Petri Nets models without the need of using the PNP methodology, which can be restrictive on the types of tasks that can be designed.

A multi-layer methodology, introduced in Costelha and Lima (2010), enables organizing separately the interaction between multiple institutions and the behavior of the robot as a single individual (which we will hereafter call “individual behavior”). While this is achieved in a higher layer, the execution of each institution can be described in a lower layer and represented on the above layer by means of macro places. By using Costelha and Lima (2010) expansion algorithm we can obtain a full Petri Net that can be tested for our desired properties. Also, this will allow us to add more



institutions on-the-fly (during the robots execution) and still maintain these properties.

## Executable Petri Nets

We follow the definitions for Petri Nets and their dynamics (enabled transitions, state transition dynamics) in Cassandras and Lafortune (2008):

*Definition:* A *Petri Net* is a five-tuple  $(P, T, A, w, X)$  where:

- $P$  is the finite set of *places*;
- $T$  is the finite set of *transitions*;
- $A \subseteq (P \times T) \cup (T \times P)$  is the set of arcs from places to transitions and from transitions to places;
- $w : A \rightarrow \mathbb{N}^+$  is the *weight function* on the arcs;
- $X$  is a *marking* of the set of places  $P$ ,  $X = [x(p_1), \dots, x(p_n)] \in \mathbb{N}^n$  represents the *state* of the Petri Net.

Herein, we assume that all the weights of the arcs are 1. If  $x(p_i)$  in marking  $X$  is equal or larger than 1, we say that place  $p_i$  is marked. Each unit in  $x(p_i)$  is called a *token*, i.e., if  $x(p_i) = 1$  then  $p_i$  has one token. State transitions in Petri Nets occur by moving tokens through the net and changing the marking by doing so. The sets of input places  $I(t_j)$  and output places  $O(t_j)$  of a transition  $t_j$  are given by  $I(t_j) = \{p_i \in P : (p_i, t_j) \in A\}$  and  $O(t_j) = \{p_i \in P : (t_j, p_i) \in A\}$ . Petri Net dynamics are provided by the following state transition function:

*Definition:* The *state transition function*,  $f : \mathbb{N}^n \times T \rightarrow \mathbb{N}^n$ , of Petri Net  $(P, T, A, w, X)$  is defined for transition  $t_j$  if and only if

$$x(p_i) \geq w(p_i, t_j) \text{ for all } p_i \in I(t_j) \quad (1)$$

If  $f(X, t_j)$  is defined, then we set  $X' = f(X, t_j)$ , where

$$x'(p_i) = x(p_i) - w(p_i, t_j) + w(t_j, p_i), \quad i = 1, \dots, n \quad (2)$$

If transition  $t_j$  verifies condition (1) then we say it is *enabled*. When transition  $t_j$  is enabled, we say that it can *fire*, and thus trigger a state change on the net by moving tokens according to (2).

Our aim is to formalize institutions as Petri Nets both for design and execution of robotic controllers. This means that we need to take into account robot actions and sensor readings. We consider three sets of building blocks that will allow us to design our controllers.

The set  $Act$  contains all robot primitive actions (combinations of two or more primitive actions are not considered as primitive actions).

The set  $Cdt$  contains boolean conditions that can be verified by checking sensor readings.

Finally, the set  $Pac$  contains “parameter actions”, which are auxiliary actions not concerning actuators but that only modify variables needed for the actions in  $Act$ .

We are now able to define our own version of Petri Nets used for execution of our robotic controllers.

*Definition:* An *Executable Petri Net* (EPN) is a Petri Net  $(P, T, A, w, X)$  where:

- each place  $p_i \in P$  has an associated action  $a_i \in Act$ ;
- each transition  $t_i \in T$  has an associated condition  $c_i \in Cdt$  and an associated parameter action  $pa_i \in Pac$ .

The basic intuition behind this definition is that by associating actions with places we are able to define which actions are to be executed at each time step. This is done simply by checking if the corresponding place is marked. By associating transitions with conditions verified by sensor readings we trigger state changes in the Petri Net due to changes in the robots environment. The following algorithm is performed by the robots at each time step, allowing the robots to execute the behavior designed in an EPN.

---

### Algorithm 1 Execute Petri Net

---

```

1: repeat
2:   for all enabled transitions  $t_i \in T$  do
3:     if associated condition  $c_i$  is true then
4:       run associated parameter action  $pa_i$ 
5:       fire transition  $t_i$ 
6:     end if
7:   end for
8: until no transition has fired
9: for all marked places  $p_i \in P$  do
10:  run associated action  $a_i$ 
11: end for
```

---

The implementation code for actions and conditions present in the sets  $Act$ ,  $Cdt$  and  $Pac$  is not explicitly represented in the code that specifies an EPN. All robots share a common function table that implements all possible actions and conditions. These are then represented in the EPN by means of indices. This allows the EPNs to be generic, in a sense that although robots may have different implementations for the same action (e.g., heterogenous robots in terms of hardware), the same EPN could be used to achieve coordination in the same manner. Also, it enables the sharing of EPNs among robots without the sharing of the actual implementation of actions.

## Institutional Agent Controller

Our goal is to formalize institutions as coordination artifacts in a modular fashion. We intend to have each institution represented by an EPN that can be executed independently or together with other institutions. The *individual behavior* for

the robots is also represented by an EPN. While the institutions specify behaviors that have a *social nature*, i.e., they relate the robot to other robots in some way, the individual behavior specifies a set of basic behaviors that have exclusively an *individual nature*, i.e., they relate the robot with the surrounding environment. The composition of the individual behavior with a set of institutions will generate a robot controller.

We now present our formalized definition of institution:

*Definition:* An *Institution*  $I$  is a four-tuple  $(Inst, initial_I, final_I, d_I)$  where:

- $Inst$  is an EPN;
- $initial_I, final_I \in Cdt$  are initial and final conditions for the execution of  $Inst$ ;
- $d_I \in D$  is the associated deontic operator.

The EPN  $Inst$  specifies the desired behavior that should be performed by the robot. This behavior is not always being executed, its start and finish are dictated by conditions  $initial_I$  and  $final_I$ , which the robot verifies at each time step. Thus, we say that an institution  $I$  at each time step can be *active* or *idle*. Each institution also includes a deontic operator  $d_I$  which is used when combining it with the robot individual behavior and further institutions. Despite  $Inst$  being designed by hand, institutions can be kept simple (e.g., arc weights set to 1) and further behavioral complexity can be reached by composition, in a modular fashion.

A previous abstract definition of institution was presented in Silva et al. (2008). There, the authors define the institution as a tuple  $(ID, Rationale, Modifiers, Network, Institutional Building, History)$ , where each element of the tuple tries to capture the main constitutive elements of the social order dynamics. For our purpose of formalizing institutions using an abstract representation, allowing for a standard design and execution platform, this definition is not sufficient. However, the EPN  $Inst$  can be seen as part of *Rationale*, since it specifies the activity of the institution, and the deontic operator as part of *Network*, since it specifies how the institution relates to other institutions.

The composition of the individual behavior with a set of institutions is non-trivial since concurrent execution of some of the institutions might be impossible or at least inadequate to the task the robot is carrying out. An example of such institutional interplay is that an institution stating that you must drive on the right side of the road will be overruled by the institution of the road code of Great Britain, and thus should not be executed when in that territory. Crawford and Ostrom (1995) define a set of deontic operators,  $D = \{P, O, F\}$ , establishing permitted ( $P$ ), obliged ( $O$ ), and forbidden ( $F$ ) operations, to be applied to institutional statements in order to deal with this problem. In our formalization, these operators

affect whether institutions are active or idle at each time step. However, the conditions that govern when a specific institution is active might refer directly to the activity state of other institutions. For instance, the institution for driving on the right is forbidden (and thus should be idle) when the institution of the road code of Great Britain is active. This referencing of other institutions creates a problem for our intended modular approach to formalization. Therefore, we have chosen to use a more restrictive set of deontic operators in order to guarantee that institutions do not refer to any other specific institution but can still prevent the concurrent execution of undesired behaviors (individual behavior and other institutions in general).

*Definition:* The set  $D$  of deontic operators for IR institutions includes the following deontic operators:  $\{AllowAll, StopInd, StopInst, StopAll\}$ . Their corresponding definitions are as follows:

- *AllowAll* implies that the associated institution can be executed concurrently with the individual behavior and all the other institutions;
- *StopInd* implies that the associated institution cannot be executed concurrently with the individual behavior;
- *StopInst* implies that the associated institution cannot be executed concurrently with other institutions;
- *StopAll* implies that the associated institution cannot be executed concurrently with the individual behavior or other institutions.

Herein we define the individual behavior simply as an EPN  $Ind$ .

As previously mentioned, Petri Nets (and thus EPN) can be represented by macro places in a hierarchical fashion, using two distinct layers. We consider that individual behavior and institutions are part of a lower layer and are represented by one macro place in the higher layer, as shown in Fig. 1. On the left side (lower layer) the EPN  $Inst$  of institution  $I$  is displayed. On the right side (higher layer) the macro place  $m_I$  representing institution  $I$  is displayed. By adding arcs from each transition in  $Inst$  to  $m_I$  and from  $m_I$  to each transition (shown as a single bidirectional dotted arc), we guarantee that each transition will only be enabled if  $m_I$  is marked. When a transition in  $Inst$  fires,  $m_I$  will continue to be marked since it is an output place of the transition.

Thus, if a macro place is marked, the individual behavior or institution that it represents is active, otherwise it is idle. This allows us to compose our institutions in the higher layer where relationships among the institutions and the individual behavior should be specified, while keeping relationships between actions and conditions separated in the lower layer. Both layers can be then merged algorithmically

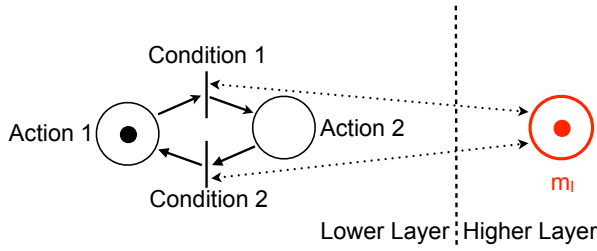


Figure 1: Hierarchical representation of an EPN in two layers. Dotted arcs represent two directional arcs, one from a transition to a place and one from a place to a transition. Left side: lower layer, EPN  $Inst$  with conditions and actions associated to transitions and places. Right side: higher layer, macro place  $m_i$  in red.

(Costelha and Lima (2010)) to obtain a full EPN that can be used as controller.

To understand how the composition of institutions is made, we consider a minimal setup with two institutions  $I_1$  and  $I_2$  and an individual behavior  $Ind$ . A representation of the higher layer of this setup before composition is presented in Fig. 2-(a). Places in red ( $m_{I_1}$ ,  $m_{I_2}$ ,  $m_{Ind}$ ) represent in the higher layer institutions ( $I_1$ ,  $I_2$ ) and the individual behavior ( $Ind$ ) implemented at the lower layer. Places  $idle_{I_1}$  and  $idle_{I_2}$  further represent the idea that institution  $I_i$  is active if place  $m_i$  is marked. Since only one place from the set  $m_i$  and  $idle_i$  can be marked at each time, we have that institution  $I_i$  is active if  $m_i$  is marked and idle if  $idle_i$  is marked. This allows us to regulate the activation and idling of institutions with their initial and final conditions as shown in the Fig. 2-(a). The individual behavior does not have an idle place since it has no initial or final conditions.

The composition of individual behavior and institutions is controlled by the deontic operators as presented in Fig. 2. As stated before, composition takes places only in the higher layer. We will see how different deontic operators for institution  $I_1$  control the composition while always maintaining the deontic operator of institution  $I_2$  as *AllowAll*. If the deontic operator of institution  $I_1$  is also *AllowAll* (Fig. 2-(a)), then no other relationship is necessary since all behaviors can be executed concurrently. If the deontic operator of  $I_1$  is *StopInst*, the structure in Fig. 2-(b) is added. Place  $idle_{Ind,I_1}$  represents the individual behavior being idle because of institution  $I_1$  being active. The added transitions have associated a special condition that is always true. This specifies that if institution  $I_1$  is activated, then the individual behavior is set to idle and vice-versa. If the deontic operator of  $I_1$  is *StopInst*, as in Fig. 2-(c), the same structure is added but now related to the macro places of the other institution and not the individual behavior. Our setup considers only two institutions but the structure would be added for

all institutions except  $I_1$ , if more institutions were present. This means that institution  $I_2$  can be idle if place  $idle_{I_2}$  is marked or if place  $idle_{I_2,I_1}$  is marked. On the latter case, institution  $I_2$  will resume being active when institution  $I_1$  becomes idle. If the deontic operator is *StopAll* then we consider a combination of the previous two cases, as show in Fig. 2-(d). These rules also apply for institution  $I_2$  if it has a different deontic operator than *AllowAll*.

We can now define our Institutional Agent Controller that will guide the performance of our robots:

*Definition: An Institutional Agent Controller (IAC) is an EPN resulting from the composition of an individual behavior  $Ind$  and a set of institutions  $\{I_1, \dots, I_n\}$  controlled by the deontic operators  $d_{I_1}, \dots, d_{I_n}$ .*

All macro places and control places ( $idle_i$ ) added during composition are associated with a void action. Considering these associations, our IAC is itself an EPN and can be executed by Algorithm 1. A minor change is needed to line 9 of the algorithm to make sure that not only the lower layer place is marked but also the higher layer macro place of the institution being executed. Time needed for the formalization includes the design time of the institutions and individual behavior and composition time. While the latter is performed algorithmically with negligible time, the former requires a certain amount of time and experience with design of behavior-based controllers (the same as with FSA).

The IAC for a desired task can be obtained prior to an experiment and transmitted to the robots. It is also possible for each robot to obtain the IAC from a given set of institutions at the start of the experiment. Thus, the method is fully scalable to any number of robots. Complexity of the IAC increases only with the number of institutions.

## Wireless Connected Swarm Case study

In this section we present a case study to illustrate how to apply our formalism of institutions in order to obtain an IAC that performs the desired task. Our aim is to be able to specify behaviors that have a social nature as institutions and behaviors that have an individual nature as individual behavior.

We have selected a case study previously investigated by Nembrini et al. (2002) and Winfield et al. (2008), where a decentralized control algorithm is able to maintain a certain degree of spatial compactness of a robotic swarm (with  $N$  robots) using exclusively, as information at the robot level, the current number of wireless connections to the neighbors. The communication is local and its bounded range a parameter of the robotic system. Let  $X$  be the number of connections perceived by a robot. In the default state, the robot simply moves forward. If at any time  $X$  falls below a threshold  $\alpha$  (where  $\alpha \in \{0, \dots, N - 1\}$ ), the robot assumes it is going in the wrong direction and turns back. Upon  $X$  returning to a value above  $\alpha$ , the robot performs a random turn and

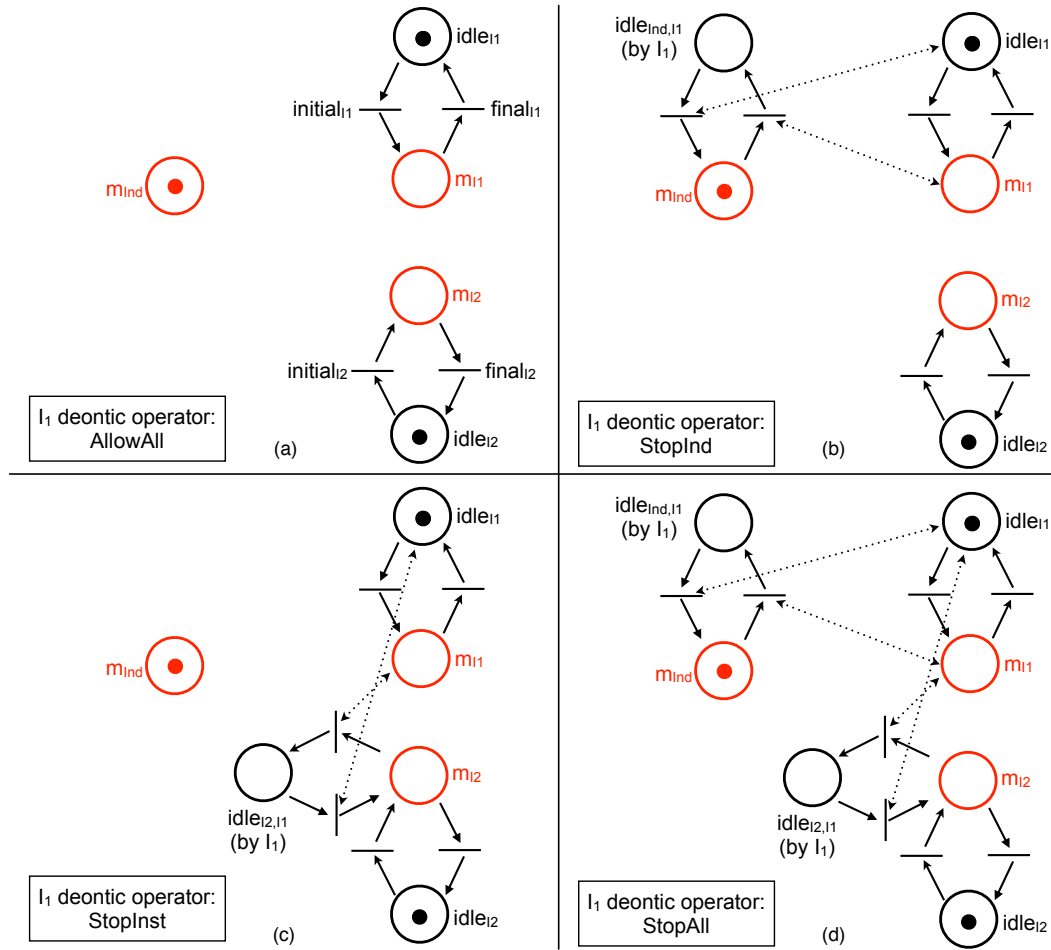


Figure 2: Composition scheme for two institutions  $I_1$ ,  $I_2$  and individual behavior  $Ind$ . Dotted arcs represent bidirectional arcs, as in Fig. 1. Places in red are macro places representing implementations of institutions and the individual behavior in the lower layer. These representations will be used throughout the paper. (a) composition rule with deontic operator *AllowAll*; (b) composition rule with deontic operator *StopInd*; (c) composition rule with deontic operator *StopInst*; (d) composition rule with deontic operator *StopAll*.

moves back to the default state. Robots always execute obstacle avoidance at the same time. This simple algorithm is quite fragile but allows the swarm to maintain its connectivity to a certain extent, with its spatial compactness being controlled by the communication range.

Our case study is similar to that of Nembrini et al. (2002) with the following differences: (i) no random turn is executed when the robots are connected again; (ii) our arena is bounded by a wall. Robots execute an individual behavior  $Ind$  and an institution  $I$ , both specified by EPNs with only two places shown in the left side (lower layer) of Fig. 3. Individual behavior  $Ind$  consists of a simple obstacle avoidance. Robots move forward until they find an obstacle (wall or other robot), perform a turn with random degree and return to moving forward. Institution  $I$  implements the social

rule, specifying that when a robot loses connections below  $\alpha$  it should turn back.

To consider the institution as defined in Section 3, we need initial and final conditions and a deontic operator. We say that initial condition  $initial_I$  is “number of connections is less than  $\alpha$ ” and the final condition  $final_I$  is “turn 180° procedure has ended”. The associated deontic operator is *StopInd* specifying that institution and individual behavior cannot be executed concurrently.

We now have all the elements needed to obtain the IAC that specifies our desired behavior. The composition of the individual behavior  $Ind$  and institution  $I$  on the left side (lower layer) of Fig. 3 is shown in the right side (higher layer) of Fig. 3. The final controller is the full EPN of Fig. 3 after the merging of the two layers. Lower layer actions and

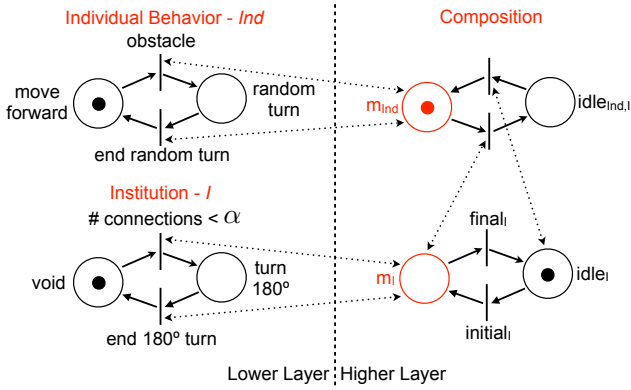


Figure 3: IAC for wireless connected swarm. Left side: lower layer EPNs for individual behavior *Ind* and institution *I*. Right side: EPN resulting from composition of individual behavior *Ind* and institution *I*.

conditions are implemented in the robot. Thus, to perform the task the robot needs only to execute Algorithm 1 taking the IAC as input. Actions associated with marked places are executed, much in the same manner as in a FSA actions associated with states would be executed.

### Corridor Case Study

A previous study concerning the institutional approach was presented in Pereira et al. (2010). Therein, institutional robotics concepts were taken into account when developing a controller for robots that had to coordinate their movement in order to traverse a narrow corridor while performing a simple transportation task. However, no formalization of the IR approach was proposed in that study. Again, our aim is to specify behaviors that have a social nature as institutions and summarize behaviors that have an individual nature as the robots' individual behavior. Our setup will consider two institutions and the individual behavior. As this case study is of higher complexity than the previous one, due to space limitations, we will not be able to describe the EPN implementations in its completeness. Therefore, we will focus only on the higher layer of the IAC.

The task consists of transporting a virtual payload in an arena with two rooms connected by a corridor. Navigation of the robots is done by performing a wall-following behavior. Transporting robots pick up the virtual payload in the left room. They must then navigate through the corridor and deploy the payload in the right room. This is the individual behavior *Ind* of the robots.

The corridor connecting the rooms is too narrow for two robots moving in opposite directions to pass one another. Thus, the robots must traverse the corridor in one direction at a time. Robots need to cooperate to avoid collisions and deadlocks in the corridor. In order to facilitate coordination,

we let a subset of the robots adopt the institutional role of “traffic regulators” to control the circulation of the remaining robots in the team. The overall traffic regulation implies robots serving as regulators and robots accepting to give priority to others in case the regulators will ask them to do so. We will therefore need two institutions, one to manage the allocation and execution of the role of regulator, and one to receive information about priority from the regulators.

If the need of traffic regulating robots arises due to a physical conflict between two robots in the corridor, these very same robots assume the role as traffic regulators. The two traffic regulators place themselves at the opposite ends of the corridor so that each regulator can control the flow of transporting robots entering the corridor from one of the rooms. The goal of the regulators is to ensure that robots only move through the corridor in one direction at a time. The regulating robots are synchronized so that only one of them will let transporting robots enter the corridor from their respective rooms at any given time. The regulation is performed by sending stop and go messages to the transporting robots.

This is clearly a behavior that has a social nature. We consider that this behavior corresponds to an institution  $I_R$  that manages the role of traffic regulator. Its initial condition  $initial_R$  is the detection of a conflict in the corridor and its final condition  $final_R$  is the end of regulation (time limit). Since we do not want this behavior to be executed concurrently with any other behavior, the deontic operator of institution  $I_R$  will be *StopAll*.

If a transporting robot receives a message to stop, it will stop in order to give priority to the robots traversing the corridor from the opposite direction. It will also begin to relay the stop message so other transporting robots behind it will stop too. As a result, the transporting robots will form a queue. When a robot in the queue receives a message to proceed, it forwards the message to any robots that may be behind it. After receiving and relaying the message the robot has priority and will traverse the corridor.

This is again a behavior that has a social nature. The behavior corresponds to an institution  $I_M$  that manages the reception and relay of messages. Its initial condition  $initial_M$  is the reception of a stop message and its final condition  $final_M$  is the reception of a go message. We do not want this behavior to be executed concurrently with the individual behavior, so its deontic operator will be *StopInd*.

In Fig. 4 we show the result of the composition of our two institutions and individual behavior. The IAC for this case study will be the result of merging this EPN with those on the lower layer.

### Conclusion and Future Work

In this work we introduced an extension to the Petri Net formalism, Executable Petri Nets. These EPN have associated actions and conditions that allow them to be executed in robots through an algorithm presented in the paper. We de-

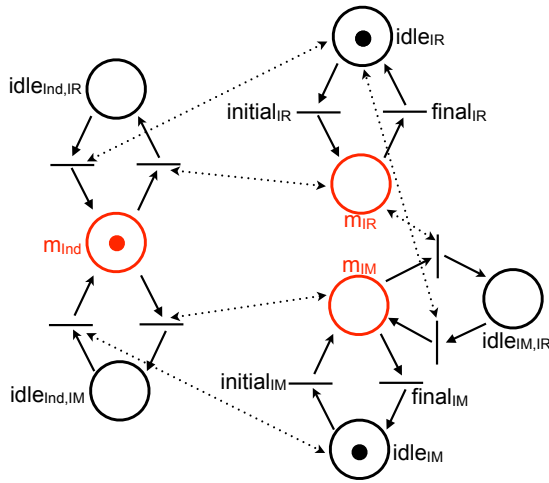


Figure 4: Higher layer EPN for corridor study. Place  $m_{Ind}$  represents the individual behavior  $Ind$ . Place  $m_{IR}$  represents institution  $I_R$ . Place  $m_{IM}$  represents institution  $I_M$ .

defined institutions and an individual behavior for robots in a distributed robotic system making use of this new extension. In our approach, institutions are modular behaviors that can be specified through an EPN and executed in a robot. Using a composition scheme controlled by dedicated deontic operators of a set of institutions we are able to obtain an Institutional Agent Controller (IAC) in the form of an EPN that combines several institutions and an individual behavior.

We applied this formalism to a simple case study where robots have to maintain wireless connections with their neighbors. We also applied the formalism to a more complex case study dealing with institutional concepts, in this case, the institutional role.

In the future we wish to study how our formalism of institutions with EPN allows us to study logical properties of the controller, such as safeness and liveness. We are also interested in studying stochastic properties of the controller, such as the steady state distribution of a given EPN or throughput of transitions. To enable this study we need to further refine our formalism of institutions to allow for stochastically timed transitions. We will also study the possibility of using the IAC as a starting point for the application of a multi-level modeling methodology. Learning of institutions and corresponding EPN will also be studied.

### Acknowledgments

This work was partially supported by Fundação para a Ciência e a Tecnologia (FCT) through grants SFRH/BD/33671/2009 (first author) as part of the Joint Doctoral Program IST-EPFL and SFRH/BPD/35862/2007 (second author), as well as by FCT ISR/IST Pluriannual funding through the PIDDAC program funds.

### References

- Cassandras, C. G. and Lafortune, S. (2008). *Introduction to Discrete Event Systems*. Springer, second edition.
- Costelha, H. and Lima, P. U. (2010). *Petri Net Robotic Task Plan Representation : Modelling , Analysis and Execution*, pages 65–90. Austria, intech edition.
- Crawford, S. E. S. and Ostrom, E. (1995). A Grammar of Institutions. *The American Political Science Review*, 89(3):582–600.
- Dias, M., Zlot, R., Kalra, N., and Stentz, A. (2006). Market-Based Multirobot Coordination: A Survey and Analysis. *Proceedings of the IEEE (Special Issue on Multirobot Coordination)*, 94(7):1257–1270.
- Hodgson, G. M. (2000). What Is the Essence of Institutional Economics ? *Journal of Economic Issues*, 34(2):317–329.
- Nembrini, J., Winfield, A. F. T., and Melhuish, C. (2002). Minimalist Coherent Swarming of Wireless Networked Autonomous Mobile Robots. *Proceedings of the seventh international conference on simulation of adaptive behavior on From animals to animats*, pages 273–282.
- Omicini, A., Ricci, A., Viroli, M., Castelfranchi, C., and Tummolini, L. (2004). Coordination artifacts: Environment-based coordination for intelligent agents. In *3rd international Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, pages 286–293, New York, NY, USA.
- Ostrom, E. (2005). *Understanding Institutional Diversity*. Princeton University Press.
- Pereira, J. N., Christensen, A. L., Silva, P., and Lima, P. U. (2010). Coordination Through Institutional Roles in Robot Collectives. In van Der Hoek, Kaminka, Lépérance, Luck, S. e., editor, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems*, pages 1507–1508, Toronto, Canada.
- Ricci, A., Viroli, M., Mater, A., and Università, S. (2005). Coordination Artifacts : A Unifying Abstraction for Engineering Environment-Mediated Coordination in MAS. *Informatica*, 29:433–443.
- Silva, P. and Lima, P. U. (2007). Institutional Robotics. In *Proc. of ECAL 2007 - 9th European Conference on Artificial Life*, pages 157–164, Lisboa, Portugal.
- Silva, P., Ventura, R., and Lima, P. U. (2008). Institutional environments. In *Proc. of Workshop AT2AI: From agent theory to agent implementation, AAMAS 2008 - 7th International Conference on Autonomous Agents and Multiagent Systems*, volume 35, pages 595–604, Estoril, Portugal.
- Tummolini, L. and Castelfranchi, C. (2006). The cognitive and behavioral mediation of institutions: Towards an account of institutional actions. *Cognitive Systems Research*, 7(2-3):307–323.
- Winfield, A. F. T., Liu, W., Nembrini, J., and Martinoli, A. (2008). Modelling a wireless connected swarm of mobile robots. *Swarm Intelligence*, 2(2-4):241–266.
- Ziparo, V. A., Iocchi, L., Lima, P. U., Nardi, D., and Palamara, P. F. (2010). Petri Net Plans: A Framework for Collaboration and Coordination in Multi-Robot Systems. *Journal of Autonomous Agents and Multi-Agent Systems*, pages 1–40.



# Coordination Games on Small-Worlds: Artificial Agents Vs. Experiments

Enea Pestelacci<sup>1</sup>, Marco Tomassini<sup>1</sup>, and Alberto Antonioni<sup>1</sup>

<sup>1</sup>Information Systems Department, HEC, University of Lausanne, Switzerland  
alberto.antonioni@unil.ch

## Abstract

Effective coordination is a key social ingredient and social structure may be approximated by networks of contacts. Using Stag Hunt games, which provide socially efficient and inefficient equilibria, we compare our simulation results using artificial players and evolutionary game theory with laboratory experimental work with human subjects on small-world type networks and with theoretical results. The conclusion is that the apparently encouraging results obtained in the few human experiments in which the local interaction structure seems to promote efficient equilibria, is neither supported by simulation results nor by theoretical ones.

## Introduction

Many types of conflicting interactions between agents in biology and society can be usefully described with the tools of Game Theory (Vega-Redondo, 2003). The Prisoner's Dilemma and the Hawk-Dove games are well known metaphor for representing the tension that appears in society when individual objectives are in conflict with socially desirable outcomes, and most of the vast research literature has focused on conflicting situations in order to uncover the mechanisms that could lead to cooperation instead of socially harmful interactions. However, in many important situations in society agents are not required to use aggressive strategies. In fact, many frequent social and economic activities simply require individuals to coordinate their actions on a common goal since in many cases the best course of action is to conform to the standard behavior. For example, if one is used to drive on the right side of the road and travels to a country where the norm is reversed, it pays off to follow the local norm. Games that express this extremely common kind of interactions are called *coordination games*.

Coordination games are apparently simple but they confront the players with multiple Nash equilibria (NE) and thus with the problem of how to choose among them. Evolutionary game theory (EGT) offers a dynamical view which is based on concepts of positively selecting fitter variants in the population, i.e. strategies that score best are more likely to survive and provides a justification for the appearance of stable states of the dynamics that represent solutions of the

game (Vega-Redondo, 2003).

For mathematical convenience, standard EGT is based on infinite mixing populations where pairs of individuals are drawn uniformly at random at each step and play the game. Correlations are absent by definition and the population has an homogeneous structure. However, everyday observation tells us that in animal and human societies, individuals usually tend to interact more often with some specified subset of partners; for instance, teenagers tend to adopt the fashions of their close friends group; closely connected groups usually follow the same religion, and so on. In short, social interaction is mediated by networks, in which vertices identify people, firms etc., and edges identify some kind of relation between the concerned vertices such as friendship, collaboration, economic exchange and so on. Thus, locality of interaction plays an important role. Recently, in the wake of a surge of activity in network research in many fields (Newman, 2003), the dynamical behavior of games on networks that are more likely to represent actual social interactions than regular grids has been investigated (see (Szabó and Fáth, 2007; Roca et al., 2009) for comprehensive recent reviews). These studies have been conducted on games of conflict such as the Prisoner's dilemma or the Hawk-Dove in most cases and have shown that there are network structures, such as scale-free and actual social networks that may favor the emergence of cooperation with respect to the fully mixing populations used in the theory (Santos et al., 2006; Luthi et al., 2008; Roca et al., 2009). Recently, some work has been done following this approach for games of the coordination type too to try to unravel the effect of structure on the population behavior, e.g. (Roca et al., 2009; Tomassini and Pestelacci, 2010).

Several analytically rigorous results are available for coordination games in well-mixed populations (Kandori et al., 1993), as well as populations with a simple local interaction structure such as rings and grids (Ellison, 1993; Morris, 2000). These results are very useful; however, while game theory has normative value, its prescriptions are not always reflected in the way people act when confronted with these situations. This has been made manifest by a host of results



of experiments with people (Camerer, 2003). Coordination games are no exception and also confront the theory with many puzzles. For coordination games on small-worlds and regular networks the recent laboratory experiments carried out in (Cassar, 2007) and in (My et al., 1999; Keser et al., 1998) are particularly relevant.

It has been argued that multi-agent learning simulations have the potential for greatly improving our knowledge of the game-theoretical interactions in artificial societies (Shoham et al., 2007). We also believe that numerical simulations, with their possibility of modeling many different situations, may shed light on the factors, both endogenous such as strategy update policy and exogenous, such as population structure, that have an influence on the game outcome. In this way, this can be a valuable tool to experiment in both the theoretical and experimental sides and to build a bridge between the two.

The paper is organized as follows. In the next section we present a brief introduction to the subject of coordination games. Then we describe the dynamical model, as well as the main results obtained in previous work. The following sections deal with the main theme of the present study, namely, the relationship between recent experimental results and our simulations. Finally, we present our conclusions.

## Coordination Games

General two-person, two strategies coordination games have the normal form of Table 1. With  $a > d$  and  $b > c$ ,  $(\alpha, \alpha)$  and  $(\beta, \beta)$  are both Nash equilibria. Now, if we assume that  $a > b$  and  $(a - d) \leq (b - c)$  then  $(\beta, \beta)$  is the risk-dominant equilibrium, while  $(\alpha, \alpha)$  is the Pareto-dominant one (Harsanyi and Selten, 1988). This simply means that players get a higher payoff by coordinating on  $(\alpha, \alpha)$  but they risk less by using strategy  $\beta$  instead. There is also a third equilibrium in mixed strategies but it is evolutionarily unstable. A well known example of games of this

	$\alpha$	$\beta$
$\alpha$	$a, a$	$c, d$
$\beta$	$d, c$	$b, b$

Table 1: A general two-person, two strategies coordination game.

type are the so-called Stag-Hunt games (Skyrms, 2004). This class of games has been extensively studied analytically in an evolutionary setting (Kandori et al., 1993; Ellison, 1993) and by numerical simulation on several model network types (Skyrms, 2004; Luthi et al., 2008; Roca et al., 2009).

## Evolutionary Games on Networks

The network of agents will be represented by an undirected graph  $G(V, E)$ , where the set of vertices  $V$  represents the

agents, while the set of edges (or links)  $E$  represents their symmetric interactions. The population size  $N$  is the cardinality of  $V$ . A neighbor of an agent  $i$  is any other agent  $j$  at distance one from  $i$ . The set of neighbors of  $i$  is called  $V_i$  and its cardinality is the degree  $k_i$  of vertex  $i \in V$ . The average degree of the network is called  $\bar{k}$ .

## Strategy Revision Rules

Since we shall adopt an evolutionary approach, we must define the decision rules by which individuals will update their strategy during time. Let  $\sigma_i \in \{\alpha, \beta\}$  be the current strategy of player  $i$  and let us call  $M$  the payoff matrix of the game, see Table 1. The quantity

$$\Pi_i(t) = \sum_{j \in V_i} \sigma_i(t) M \sigma_j^T(t)$$

is the accumulated payoff collected by agent  $i$  at time step  $t$  and  $\sigma_i(t)$  is a vector giving the strategy profile at time  $t$ .

Here we shall describe two among the most commonly used strategy revision rules. These rules, although they are extremely simple, also make sense when human players are concerned, at least at a very low level of knowledge and information processing capabilities. The first rule is to switch to the strategy of the neighbor that has scored best in the last time step. This *imitation of the best* policy can be described in the following way: the strategy  $\sigma_i(t)$  of individual  $i$  at time step  $t$  will be

$$\sigma_i(t) = \sigma_j(t-1),$$

where

$$j \in \{V_i \cup i\} \text{ s.t. } \Pi_j = \max\{\Pi_k(t-1)\}, \forall k \in \{V_i \cup i\}.$$

That is, individual  $i$  will adopt the strategy of the player with the highest payoff among its neighbors including itself. If there is a tie, the winner individual is chosen uniformly at random, but otherwise the rule is deterministic.

At a slightly higher sophistication level, a well known adaptive learning rule is *myopic best-response* (Young, 1998), also called best-reply, which embodies a primitive form of bounded rationality and for which rigorous results are known. In the local version of this model, time is discrete i.e.  $t = 0, 1, 2, \dots$  and, at each time step, an agent has the opportunity of revising her current strategy with probability  $p$ . She does so by considering the current actions of her neighbors and switching to the action that would maximize her payoff if the neighbors would stick to their current choices. In other words,  $\hat{\sigma}_i$  is a best response for player  $i$  if  $\Pi_i(\hat{\sigma}_i(t)) > \Pi_i(\sigma_i(t)), \forall \sigma_i$ . In case of a tie, agent  $i$  keeps its current strategy.

The model is thus completely local and an agent only needs to know her own current strategy, the game payoff matrix, who are her neighbors, and their current strategies. This

rule is called myopic because the agents only care about immediate payoff, they cannot see far into the future. In order to introduce some stochasticity, an agent can make a mistake with some small probability  $q$ . These small random effects are meant to capture various sources of uncertainty such as deliberate and involuntary decision errors. Deliberate errors might play the role of experimentation, and involuntary ones might be linked with insufficient familiarity with the game, for example. This dynamic will be called best-response with noise.

The simulation represents a dynamical system in which time  $t$  is discrete, i.e.  $t = 0, 1, \dots$ . Let us call  $\Sigma(t) = (\sigma_1(t), \dots, \sigma_N(t))$  the strategy profile at time  $t$ . For the imitation of the best and best response rules the evolution of  $\Sigma(t)$  is deterministic. In the best response with noise case the resulting process is stochastic. It can be described by a Markov chain (Kandori et al., 1993) since the probability of strategy profile  $\Sigma(t) = (\sigma_1(t), \dots, \sigma_N(t))$  at time step  $t + 1$  only depends on the previous time step:

$$Pr(\Sigma(t+1) | \Sigma(t), \Sigma(t-1), \dots) = Pr(\Sigma(t+1) | \Sigma(t)).$$

It is clear that more refined forms of learning, such as reinforcement learning could be used to represent the agents' decisions (Camerer, 2003). However, these more sophisticated approach do not have yet a firm theoretical basis and could not be compared with baseline dynamical models. This is the reason why, in the interest of simplicity, we stick with very simple basic protocol revision rules here.

## Summary of Previous Simulation and Theoretical Results

This section summarizes previous numerical results on Stag Hunt games. Several populations topologies have been studied, including regular lattices (Skyrms, 2004; Roca et al., 2009), random graphs (Roca et al., 2009; Luthi et al., 2008), scale-free graphs (Luthi et al., 2008; Roca et al., 2009), model and actual social networks (Luthi et al., 2008) using several strategy update rules such as replicator dynamics, imitation of the best, and best response dynamics. In the average, for initially equidistributed strategies, at the steady state the population is monomorphic, with all individuals playing  $\alpha$  or  $\beta$ . For all network types, the more efficient  $\alpha$  strategy is enhanced with respect to what would happen in a mixing population. This is true for all update rules except best reply, for which the topology does not seem to play an important role (Roca et al., 2009). Social networks also favor the Pareto-efficient outcome in the average but the steady state population is often dimorphic, i.e. there is a mix of the two strategies. The reason why there can be mixed states in social networks has been attributed (Tomassini and Pestelacci, 2010) to the presence of *communities*. In fact, social networks can usually be partitioned into recognizable clusters (Newman, 2003); within these clusters strategies

may become dominant as in the pure coordination case just by chance. In other words, as soon as a strategy dominates in a given cluster, it is difficult to eradicate it from outside since other communities, being weakly connected, have little influence.

We now briefly comment on the relationship between the results of numerical simulations and well known theoretical results on Stag-Hunt games (for a recent review see (Weidenholzer, 2010)). These theoretical models are based on ergodic stochastic processes in very large well mixed populations and state that, when using best-response dynamics in random two-person encounters, and in the presence of a little amount of noise, both for well mixed populations as well as for populations structured as rings, the risk-dominant strategy should take over the population in the long run (Kandori et al., 1993; Ellison, 1993). But coordination seems to be sensitive to the exact type of revision protocol and dynamic. For example, Robson and Vega-Redondo (Robson and Vega-Redondo, 1996) found that the Pareto-dominant equilibrium is selected if players are immediately randomly rematched after each encounter.

Simulations results on networked populations indirectly confirm the above, i.e., at the steady state there is always either a single strategy, but not necessarily the risk-dominant one. However, owing to network reciprocity effects related to clustering, a mix of both strategies is also possible. In summary, it can be said that network effects tend to reinforce cooperation on the Pareto-dominant case, which is a socially appreciable effect. However, these results must be taken with a grain of salt. Numerically studies deal with finite, network-structured populations during a limited amount of time, while theoretical results have been established for large well mixed populations in the very long run. Thus, numerical results and theoretical predictions based on different assumptions do not necessarily agree with each other.

## Discussion of Some Experimental Results on Coordination Games

In this section we comment on some experimental results on coordination games in the light of the conclusions that have been reached by numerical simulation and also with respect to theoretical results. There have been many experiments in the field and we cannot be exhaustive; however, the main conclusions are the following. When the analog of a (finite and generally small) well mixed population of players have been used, the general result is that polymorphic final states are rare, the initial state of play i.e. the strategy played at the first period is a good predictor of convergence, and the risk-dominant equilibrium is often reached in the laboratory, i.e. coordination failures emerge, although in some cases, especially in finitely repeated games and by varying the payoff structure, coordination on the Pareto-efficient equilibrium can also be achieved (Cooper et al., 1992; Bat-

talio et al., 2001). It has also been observed that the number of rounds, the size of the group, and the fact of playing repeatedly with the same player may influence the result, i.e. small groups, higher number of rounds, and repeated interactions have been shown to favor the Pareto-efficient outcome (Huyck et al., 1993, 1990). This lends support to the idea that human agents play the games using some imperfect decision rules that, nonetheless, may be similar to some variant of myopic best response, perhaps with a longer memory of past encounters instead of just one step behind. No doubt, human decision-making is a lot more complex, but simple learning rules should somehow evolve during these experiments.

A more interesting situation from the point of view of the present work is the one in which some more specific population structure has been recreated in the laboratory setting. We are aware of three experiments of this type, the work of (Cassar, 2007), and the studies of (My et al., 1999) and of (Keser et al., 1998).

Keser et al. used a ring structure where each player has a neighbor on either side and a well mixed structure for comparison. Their conclusions were that in the ring the preferred equilibrium is the risk-dominant one, while the payoff-dominant equilibrium was the more frequent result in the globally communicating population. This is in qualitative agreement with the theoretical predictions of (Ellison, 1993) for the ring and (Robson and Vega-Redondo, 1996) for the mixing case.

My et al. performed a comparative experimental study of Stag Hunt games with three different payoff matrices on mixing and structured populations. The population with local structure was composed by a circle of eight people where each player only interacts with her immediate right and left neighbors. They find that the first period modal choice of strategy, which is the payoff dominant one, plays a major role in the final outcome. In the global population case, the steady state generally lies in the same basin of attraction as the initial state. This result, which is commonly observed in many laboratory experiments, does not agree with the theoretical results of (Kandori et al., 1993) which predict that all the probability at stochastic equilibrium be placed on the risk-dominant state. However, we have to bear in mind that the latter have been established for stochastic processes in the very long run taking place in large populations and neither of these conditions can be satisfied in a laboratory setting.

For the ring structure, the convergence to the risk-dominant outcome is more frequent than in the well mixed case, especially when the payoff matrix values are such that the Pareto-superior basin shrinks. However, still often times the system converges to the Pareto-dominant state, which disagrees with the theoretical predictions of (Ellison, 1993) based on noisy best reply dynamics. By examining the detailed history of play, the experimenters have found that, while in the

global population subjects on average play myopic best response, in the ring with local structure a kind of imitation rule fits the data better than best reply. This is in qualitative agreement with the very extensive numerical studies of (Roca et al., 2009), where the simple strategy of imitating the individual having the best payoff in the neighborhood is the one that best promotes cooperation in the Stag Hunt played on several classes of networks.

## The experimental Study of Cassar and its Numerical Simulation

The study of Cassar (Cassar, 2007) is the most interesting one from the standpoint of the present paper as it investigates network structures that are more realistic than the ring and the two-dimensional lattice, although the ring is also used in the experiments for comparison. One particular Stag Hunt payoff matrix is used in (Cassar, 2007) with the following payoff values (see Table 1):  $a = 5$ ,  $b = 1$ ,  $c = -1$ ,  $d = 4$ . With this choice the frequency  $\alpha$  of stag players at the (unstable) mixed equilibrium would be  $2/3$ , since at this point the expected value playing strategy  $\alpha$ ,  $E[\alpha]$ , is equal to the expected value  $E[\beta]$ , which implies  $5p - (1-p) = 4p + 1 - p$ , i.e.  $p = 2/3$ , where  $p$  is the probability with which  $\alpha$  is played in the mixed strategy or, equivalently, the  $\alpha$  fraction in the population. This leads to a basin of attraction for the payoff-dominant strategy  $\alpha$  that is half the size of the corresponding basin for the risk-dominant strategy.

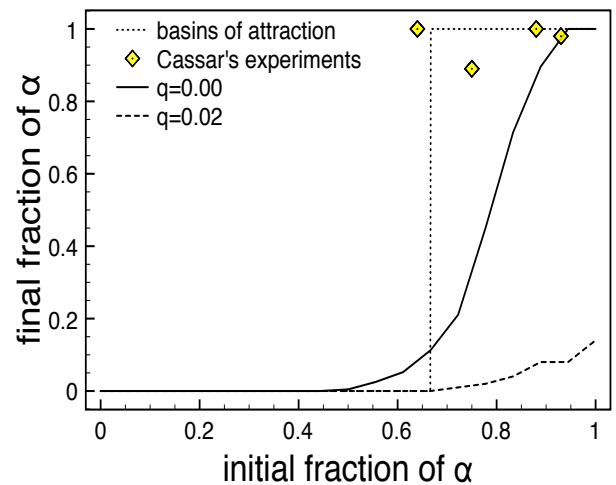


Figure 1: Final average ratio of  $\alpha$ -players as a function of their initial ratio in small-world networks of size  $N = 18$  and  $\bar{k} = 4$ . With noise (dashed curve) the system converges almost always to the risk-dominant steady state. Without noise (continuous curve) the payoff-dominant steady state is often reached when the initial ratio is in the corresponding basin of attraction. The dotted line marks the theoretical Nash equilibria and their basins of attraction. The small squares represent the results of Cassar's experiments.

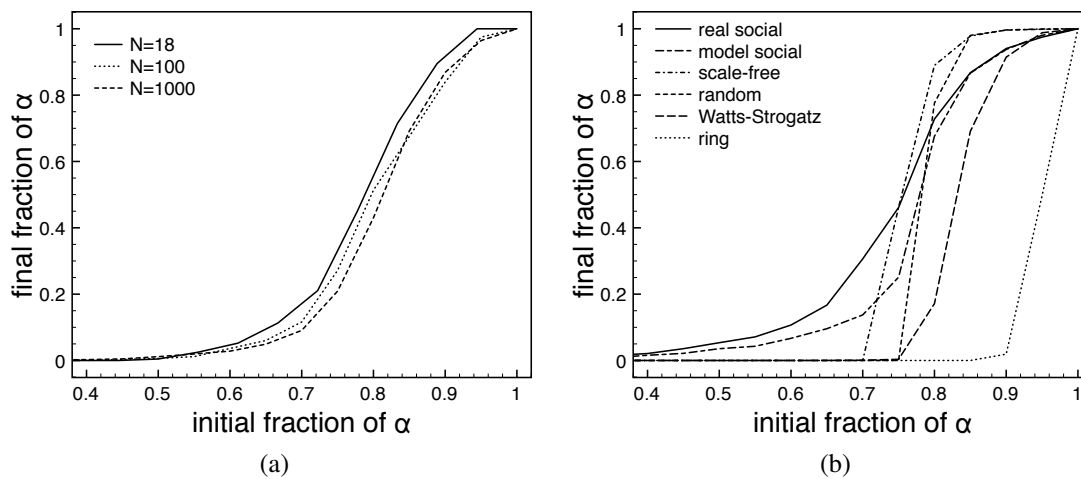


Figure 2: Final ratio of  $\alpha$ -players as a function of their initial ratio. (a): Watts–Strogatz small-world networks scaling;  $\bar{k} = 4$ . (b): Other network topologies with size  $N = 1000$  and  $\bar{k} = 6$ . The horizontal scale starts at  $\alpha = 0.4$ . Results are averages over 50 runs for each network class using myopic best response as a strategy update rule.

Summarizing Cassar’s experimental settings, groups of 18 subjects were used, virtually connected with a local network according to three types of graph topology: ring, random, and Watts–Strogatz small world (Watts and Strogatz, 1998). Watts–Strogatz graphs are constructed starting from a regular lattice of low degree and rewiring each link in turn with some small probability to a node chosen uniformly at random. Thanks to the formation of shortcuts between distant parts of the ring the clustering coefficient remains high, while the path lengths are dramatically shortened. Although the resulting graphs are poor representations of actual social networks, some statistical quantities are qualitatively correctly reproduced (Watts and Strogatz, 1998; Newman, 2003).

The degree of each node  $k$  was exactly 4 for the ring, while it was  $k = 4$  on the average for random and small-world networks. Of course, a single realization of the ring was used, while three different realizations of each of the other two topologies were generated.

Cassar’s results can be summarized as follows. In all three networks the Pareto-dominant equilibrium was the preferred result, with a significant advantage for the small-world networks in terms of coordination on the efficient outcome. Likewise, the ring was more favorable than the random graph. The frequency of choice of the Pareto-dominant outcome on the small-world graphs is unusually high, about 95%. Thus, the qualitative conclusion is that rings, and especially small-world networks, are favorable topological structures for coordination on the socially efficient outcome. This is in contrast with theoretical results on rings using noisy best response dynamics (Ellison, 1993) while there are no theoretical results on Watts–Strogatz small worlds to compare with. However, from the extensive numerical work

of (Roca et al., 2009) it appears that several different graph structures do favor the payoff-dominant equilibrium in the population for the Stag Hunt for most of the strategy update rules tried, but not for best reply dynamics. In light of the above, Cassar’s results seem to us less compelling than they would appear at first sight.

To obtain more insight into the matter, we decided to simulate the game behavior on an ensemble of computer-generated networks of the same size  $N = 18$  as those used by Cassar with best response dynamics. We are aware of the limitations of the comparison: artificial agents are not the same thing as rational or semi-rational humans in the laboratory and time scales are vastly different since only a limited number of runs can be effectively tested in experiments. Nevertheless, we think that the exercise is worthwhile and can shed some light into the question. One important thing to note is that in (Cassar, 2007) the first period move in most cases is the payoff-dominant one, which might be due to psychological reasons in human subjects and is frequently observed in experiments (see also (Battalio et al., 2001; My et al., 1999)). In order to explore the whole spectrum thus avoiding such initial bias in the simulations, we have studied several different initial proportions between 0 and 1. Small-world instances were generated anew for each run and each computed point is the average of 50 runs. We have used a fully asynchronous update scheme in which a randomly selected agent is chosen for update with replacement at each discrete time step. To detect steady states of the dynamics we first let the system evolve for a transient period of  $5000 \times N \simeq 5 \times 10^6$  time steps. After a quasi-equilibrium state is reached past the transient, averages are calculated during  $500 \times N$  additional time steps. A steady state has always been reached in all simulations performed within the

prescribed amount of time, for most of them well before the limit.

As an update rule we used both myopic best reply as well as best reply with a small amount of mutation  $q = 0.02$ . Figure 1 reports the average results of 50 runs for each case. As prescribed by theory (Kandori et al., 1993; Ellison, 1993) and confirmed by simulations, the noisy dynamics leads essentially to risk-dominant outcomes. On the other hand, with deterministic best response dynamics, the results are that, in general, the system reaches at steady state the basin corresponding to its initial strategy proportion, with a slight advantage for the risk-dominant equilibrium, also in qualitative agreement with the expected theoretical results. Focusing more specifically on the average initial conditions that arose in Cassar's experiment, i.e. with a proportion of  $\alpha$  of about 0.7, one sees that at this point the amount of cooperation found in the simulations is much lower, about 0.30 instead of full or almost full cooperation found in the few laboratory experiments shown as small squares. Again, note that the above results are for automata playing mechanically a deterministic myopic best response. Instead, Cassar's results have been obtained with human players; nevertheless, the difference is striking.

Cassar tried to relate her results to some statistical topological features of the networks. Her main suggestion was that the higher the clustering coefficient<sup>1</sup>, the higher the probability of players choosing the Pareto-superior strategy. Unfortunately, given the small size  $N = 18$  of such networks and only three network realizations each for random and Watts–Strogatz, all sampled quantities such as the degree distribution function  $p(k)$  and mean clustering coefficient  $\bar{C}$  are too noisy to be statistically significant. For example, for a random graph, the clustering coefficient  $C$  asymptotically tends to 0 as  $N \rightarrow \infty$ . However, for small  $N$  clustering remains high in random graphs, which is actually the case for the values reported by Cassar. Thus, it is difficult to relate  $\bar{C}$  with the game dynamics for such small networks. With those caveats in mind, in order to get an idea as to the effect on the dynamics of scaling-up the network, we report in Fig. 2 (a) the results on graphs of size  $N = 100$  and  $N = 1000$ , together with those for  $N = 18$ . It is apparent that, apart from smoothing the finite-size fluctuations, scaling-up the graph has only the effect of shifting the inset of cooperation on the payoff-dominant outcome a bit further to the right. In Fig. 2 (b) we report the fraction of population coordinating on the payoff-dominant strategy  $\alpha$  as a function of the initial proportion of  $\alpha$ -strategists in various network types of size  $N = 1000$  for the payoff values used in Cassar

(2007). It can be seen that the clustering coefficient does not seem to play an important role on the population behavior. In fact, rings and Watts–Strogatz small-world graphs which both have high clustering values lead to the lowest amount of payoff dominance. On the other hand, both model and real social networks, which also have high clustering, show more coordination on the payoff-dominant strategy for  $\alpha$  below the theoretical 0.66 value, as well as slightly diminished value in the region above this value. The explanation for this behavior is related to the community structure that these networks possess (Tomassini and Pestelacci, 2010). In fact, very often at steady state the population is polymorphic, with a minority of clusters in which  $\alpha$  dominates below 0.66 and a minority of clusters of agents playing  $\beta$  above this limit. Table 2 illustrates the above by giving the mean clustering coefficient  $\bar{C}$  and the modularity  $Q^2$  of the irregular network types for  $N = 1000$ . The modularity values have been computed with Newman's and Girvan's divisive algorithm based on betweenness Newman and Girvan (2004).

In conclusion, these numerical experiments confirm that the key factor to promote cooperation in networks of agents playing coordination games according to best response when risk-dominance should theoretically prevail, is the network community structure, not the clustering coefficient. Conversely, this same community structure makes it possible for a fraction of  $\beta$ -strategists to survive in clusters when payoff-dominance should prevail.

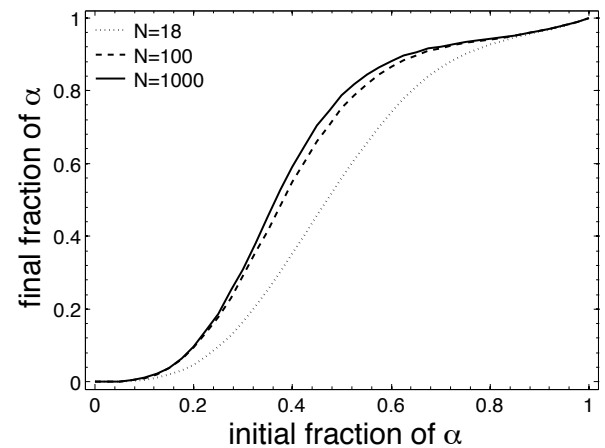


Figure 3: Final average fraction of  $\alpha$ -players as a function of their initial fraction in the population in small-world networks of size  $N = 18$ ,  $N = 100$ , and  $N = 1000$  with  $\bar{k} = 4$ . Agent strategy update rule is by imitation of the best.

<sup>1</sup>The clustering coefficient  $C_i$  of a node  $i$  is defined as  $C_i = 2E_i/k_i(k_i - 1)$ , where  $E_i$  is the number of edges in the neighborhood of  $i$ . Thus  $C_i$  characterizes the extent to which nodes adjacent to node  $i$  are connected to each other. The clustering coefficient of the graph is simply the average over all nodes:  $C = \frac{1}{N} \sum_{i=1}^N C_i$  (Newman, 2003).

<sup>2</sup>According to Newman (Newman, 2006), where quantitative definitions are given, modularity is proportional to the number of edges falling within clusters minus the expected number in an equivalent network with edges placed at random.

	Ring	Small-World	Random	Scale-Free	Model Social	Real Social
$\bar{C}$	0.6	0.44	0.006	0.03	0.57	0.69
$Q$	-	-	0.31	0.30	0.66	0.69

Table 2: Average clustering coefficient  $\bar{C}$  and modularity  $Q$  for various network types of size  $N = 1000$  and  $\bar{k} = 6$ . The values are averages over 20 independent graph realizations. A - sign means that  $Q$  is not meaningful.

Since multi-agent simulations are cheap, while laboratory experiments demand a lot of time and resources, we have also simulated the same system assuming that the agents play unconditional imitation of the best in their neighborhood, instead of playing best response. Imitation of the best is a primitive strategy for humans, but it could be used in the absence of more refined reasoning tools, as in the experiment of (My et al., 1999). After all, such imitative behavior is very common in the stock market. The results for different initial shares of  $\alpha$  and for three network sizes are shown in Fig. 3, and should be compared with Figs. 1 and 2 (a). The notable feature is that the fraction of population playing  $\alpha$  is strongly enhanced with respect to the simulations using best reply as a strategy update rule. This is in agreement with the numerical findings of (Roca et al., 2009) where it is shown that unconditional imitation of the best gives rise to the highest amount of efficient coordination on all network types tested. Indeed, Cassar’s experimental observations would be much closer to the results using imitation of the best than to those updating with best reply, as can be seen by comparing Figs. 1 and 3. However, in Cassar’s experiment, neighbors’ payoffs were not made known to the players and thus they could not employ a decision rule based on payoff differences. Indeed, Cassar’s analysis of the subjects’ behaviors favored rules based on myopic best reply and inertia, which means that after having chosen a strategy, a player may keep it for some time.

Clearly, a delicate point is the actual decision rule, or rules, humans do use during these experiments. While the simulated protocol revision rules used in simulations are extremely simple and homogeneous in the agents population, this is probably not the case with human players. Certainly, some amount of more sophisticated learning is at work which is not fully represented in the basic rules, as explained in Camerer’s book (Camerer, 2003), for example. For this reason, we think that it is extremely useful to validate statistical learning models arising from the experiments. These could then in turn guide and pave the way for better and more realistic strategy revision rules.

## Summary and Conclusions

In this work we have studied general coordination games on complex networks by numerical simulation and we have compared the results with those of the few experimental studies that have been performed on structured populations.

For general coordination games of the Stag Hunt type there is a tension between payoff-dominance and risk-dominance and thus it is of interest to know whether there exist population topology conditions that might favor the socially efficient, Pareto-superior outcome. We have simulated a particular, yet representative, coordination game on several classes of complex networks in order to compare the results with the laboratory experiments of Cassar (Cassar, 2007). This experiment with human beings is, to our knowledge, the only one to date which employs complex network structures resembling, at least from some statistical point of view, real social networks. Our results suggest that Cassar’s claims on the role of Watts–Strogatz small-world networks, and especially their clustering coefficient, on the predominance of payoff-dominant outcomes are inconclusive and are essentially due to favorable average initial conditions. These, in turn, seem to be a bias that is almost always present in such experiments and which may well be due to human psychological propensities, something that cannot be reproduced by the artificial agents used in the simulations but which can be easily simulated by generating the corresponding initial conditions. The numerical work also show that an important source of promotion of the efficient outcome is due to the community structure present in some networks for reasonable-sized networks, i.e. with a size of at least one hundred nodes. This, however, cannot be directly related to the experimental studies as the size of the populations used in the latter have been too small till now for any meaningful partition into clusters. In conclusion, we suggest that further laboratory work on a larger scale, such as those reported in Grujic et al. (2010) should be performed to elucidate the role that complex networks of contacts may have on the emergence of efficient coordination patterns when human agents are considered. In conclusion, we think that, although numerical multi-agent simulations cannot be directly compared with heterogeneous and possibly complex human decision rules, they are a useful guide for planning and interpreting laboratory experiments and social dynamics in general.

## Acknowledgements

A. Antonioni and M. Tomassini are grateful to the Swiss National Science Foundation for financial support under contract number 200021-132802/1.

## References

- Battalio, R., Samuelson, L., and Huyck, J. V. (2001). Optimization incentive and coordination failure in laboratory stag hunt games. *Econometrica*, 61:989–1018.
- Camerer, C. F. (2003). *Behavioral Game Theory*. Princeton University Press, Princeton, NJ.
- Cassar, A. (2007). Coordination and cooperation in local, random and small world networks: Experimental evidence. *Games and Economic Behavior*, 58:209–230.
- Cooper, R., DeJong, D. V., Forsythe, R., and Ross, T. W. (1992). Communication in coordination games. *Quarterly Journal of Economics*, 107:739–771.
- Ellison, G. (1993). Learning, local interaction, and coordination. *Econometrica*, 61:1047–1071.
- Grujic, J., Fosco, C., Araujo, L., Cuesta, J. A., and Sánchez, A. (2010). Social experiments in the mesoscale: humans playing a spatial Prisoner’s Dilemma. *PLoS ONE*, 5(11):e13749.
- Harsanyi, J. C. and Selten, R. (1988). *A General Theory of Equilibrium Selection in Games*. MIT Press, Cambridge, MA.
- Huyck, J. B. V., Battalio, R. C., and Beil, R. O. (1990). Tacit coordination games, strategic uncertainty, and coordination failure. *Amer. Econ. Rev.*, 80:234–249.
- Huyck, J. B. V., Battalio, R. C., and Beil, R. O. (1993). Asset markets as an equilibrium selection mechanism: Coordination failure, game form auctions, and tacit communication. *Games Econ. Behav.*, 5:485–504.
- Kandori, M., Mailath, G., and Rob, R. (1993). Learning, mutation, and long-run equilibria in games. *Econometrica*, 61:29–56.
- Keser, C., K.-M.-Erhart, and Berninghaus, S. (1998). Coordination and local interaction: experimental evidence. *Economics Letters*, 59:269–275.
- Luthi, L., Pestelacci, E., and Tomassini, M. (2008). Cooperation and community structure in social networks. *Physica A*, 387:955–966.
- Morris, S. (2000). Contagion. *Review of Economic Studies*, 67:57–78.
- My, K. B., Willinger, M., and Ziegelmeyer, A. (1999). Global versus local interaction in coordination games: an experimental investigation. Technical Report 9923, Working papers of BETA. ULP, Strasbourg.
- Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, 45:167–256.
- Newman, M. E. J. (2006). Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA*, 103:8577–8582.
- Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113.
- Robson, A. J. and Vega-Redondo, F. (1996). Efficient equilibrium selection in evolutionary games with random matching. *J. Econ. Theory*, 70:65–92.
- Roca, C. P., Cuesta, J. A., and Sánchez, A. (2009). Evolutionary game theory: temporal and spatial effects beyond replicator dynamics. *Physics of Life Reviews*, 6:208–249.
- Santos, F. C., Pacheco, J. M., and Lenaerts, T. (2006). Evolutionary dynamics of social dilemmas in structured heterogeneous populations. *Proc. Natl. Acad. Sci. USA*, 103:3490–3494.
- Shoham, Y., Powers, R., and Granager, T. (2007). If multi-agent learning is the answer, what is the question? *Artificial Intelligence*, 171:365–377.
- Skyrms, B. (2004). *The Stag Hunt and the Evolution of Social Structure*. Cambridge University Press, Cambridge, UK.
- Szabó, G. and Fáth, G. (2007). Evolutionary games on graphs. *Physics Reports*, 446:97–216.
- Tomassini, M. and Pestelacci, E. (2010). Evolution of coordination in social networks: A numerical study. *Int. J. Mod. Phys. C*, 21(10):1277–1296.
- Vega-Redondo, F. (2003). *Economics and the Theory of Games*. Cambridge University Press, Cambridge, UK.
- Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *Nature*, 393:440–442.
- Weidenholzer, S. (2010). Coordination games and local interactions: a survey of the game-theoretic literature. *Games*, 1:551–585.
- Young, H. P. (1998). *Individual Strategy and Social Structure*. Princeton University Press, Princeton.



# A DNA toolbox for engineering *in vitro* life-like behaviors

Raphal Plasson<sup>1</sup>, Kevin Montagne<sup>2</sup>, Adrien Padirac<sup>2</sup>, Teruo Fujii<sup>2</sup>, Yannick Rondelez<sup>2</sup>

<sup>1</sup>Dep. of Appl. Chem., Faculty of Science and Technology, Keio University,  
3-14-1 Hiyoshi, Kouhoku-ku, Yokohama 223-8852, Japan

<sup>2</sup>LIMMS/CNRS-IIS, Institute of Industrial Science, The University of Tokyo,  
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan  
plasson@aplc.keio.ac.jp

Living organisms perform and control complex behaviors by using webs of chemical reactions organized in precise networks. Understanding how life-like behaviors emerge from such complex chemical systems is a challenge for artificial life scientists. An approach is to implement minimal *in vitro* systems, possessing the characteristic dynamic properties of living systems. In a bottom-up perspective, the ultimate purpose is to lead to the description of minimal functional cells. Taking example on the modularity of biosystems Kitano (2002), complex artificial networks can be obtained by the assembly of elementary building blocks Qian and Winfree (2011). In that scope, we developed an experimental framework of dynamic DNA-based modules, that can be assembled to generate large networks with non-trivial dynamic.

This study focuses on the description of a minimal cell as a computing unit. With respect to their environment, simple organisms like bacteria must perform a number of basic computing operations: detection of chemical gradients (chemotaxis), prediction of night and days alternation (circadian rhythms) or remembering of past decisions. In molecular terms, these behaviors correspond to various information processing abilities, like adaptation, oscillations, or bistable switching. They are performed within the cell by networks of intercoupled biochemical reactions, one prominent example being the gene regulatory networks.

Our work consisted in building experimental chemical webs that can implement such dynamic functions. We developed a modular DNA toolbox based on a simple biochemical machinery, enabling the construction of arbitrary chemical networks, and their easy *in vitro* implementation (Montagne et al., 2011). A theoretical work was performed in a continuous feedback loop with the experimental implementation. Simulations of the chemical networks are used for their design, their optimization and their study. Based on the knowledge of the thermodynamic and kinetic parameters of individual reactions, numerical integrations of the corresponding ODE sets enable the assembly of novel networks for predicting their behavior, and to adapt the network topologies for obtaining the target behavior.

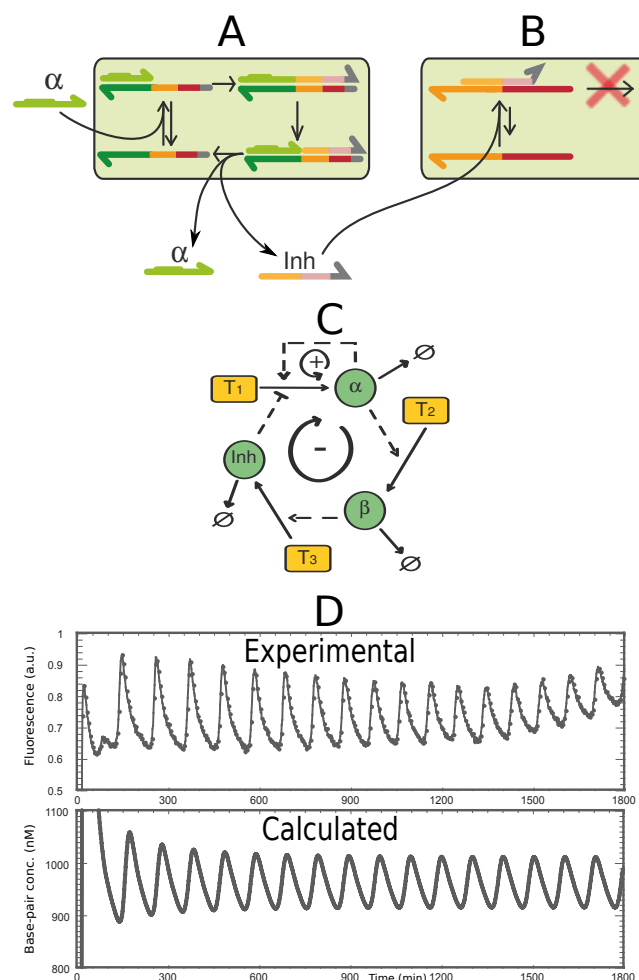


Figure 1: A DNA toolbox. An activation module (A) can be designed for synthesizing a specific oligonucleotide (Inh) when a signal oligonucleotide ( $\alpha$ ) is present. An inhibition module (B) can be built from the synthesis of an oligonucleotide (Inh) that can specifically interact with a module in order to block its activity. These modules can be assembled in a chemical oscillator (C) that was experimentally designed with a predictable behavior (D).

This system is based on the replication of DNA strands by enzymatic reactions (see Fig. 1). Template DNAs are designed for producing specific message strands, when activated by specific signal strands. Autocatalytic networks are obtained when the signal strand is identical to the message strand. An inhibitor strand can be designed for each template. Full networks can be obtained by assembling these modules, generating positive and negative feedbacks. The dynamics of the system is guaranteed by the presence of an excess of activated nucleotide monomers, and the continuous destruction of the oligomers, for sustaining reaction fluxes.

This toolbox can be used to build non-trivial behaviors. As a proof of concept, we recently reported the de novo construction of a biochemical oscillator, by assembling an autocatalytic unit with a negative feedback loop (Montagne et al., 2011). The dynamic behavior (stability, period, amplitude) of this experimental system can be quantitatively predicted and modulated. We'll discuss how the same toolbox can be used to construct other life-like functions, like bistable or gradient responsive switches, but also logical gates or boolean networks. In the future, compartmentalization of these amorphous systems in vesicles or droplets may provide a good platform for the design of autonomous protocells.

## References

- Kitano, H. (2002). Computational systems biology. *Nature*, 420(6912):206–210.
- Montagne, K., Plasson, R., Sakai, Y., Fujii, T., and Rondelez, Y. (2011). Programming an in vitro dna oscillator using a molecular networking strategy. *Mol Syst Biol*, 7:466.
- Qian, L. and Winfree, E. (2011). Scaling up digital circuit computation with dna strand displacement cascades. *Science*, 332(6034):1196–1201.

# Programs Self-Healing over a termites simulator based on language games and evolutionary computing

Arles Rodríguez Portela<sup>1,2</sup>, Jonatan Gómez Perdomo<sup>1,2</sup>

<sup>1</sup>Alife Research Group

<sup>2</sup>National University of Colombia

aerodriguezp@unal.edu.co, jgomezpe@unal.edu.co

## Abstract

This paper presents a mechanism of programs self-healing in an environment of agents looking for food. The failure system is defined based on initial failures that each agent (termite) has on their programs. By using language games concepts and the Q-learning algorithm, termites diagnose failures on their programs. Termites also have enough information to determine if their programs are failing based on a simple voting system that is the result of language games of diagnosis. The proposed self-healing mechanism was tested on virtual worlds with 100 and 200 termites and a different failure per termite. The results show that the proposed approach is capable, from local interactions, of building a set of very specific diagnosis questions, allowing the system to diagnose more than one type of failure at the same time, while the accounted number of diagnosis questions for instructions with low failure probability is reduced. By using the voting system and storing a ranking of possible missing code lines, mutations are induced on the code and the system is capable of recovering the programs.

## Introduction

Self-healing is based on the ability to detect software and hardware components that are failing. Systems must detect failures on components and then replace, eliminate or repair them without disrupting the system operation (Nami and Bertels, 2007).

Self-healing involves: the design and verification of an autonomic system which has some of the complexity of a real system in order to locate functions and services offered by an autonomic element in an efficient manner (Kephart and Chess, 2003), to make an abstraction of behaviors to obtain emergent properties and global behaviors from local actions (Kephart and Chess, 2003; Biccocchi and Zambonelli, 2007; De Wolf and Holvoet, 2003), to reallocate resources (Arora et al., 2006) and to locate faults as fast as possible (Gao et al., 2004).

An important part of the problem is to develop a virtual organization in an area where certain items may have certain types of failures and to reduce the risk of large losses by getting a reconfiguration that ensures the continuity of the

system and the potential generation of learning about corrective actions (Nami and Sharifi, 2007; Gao et al., 2004).

Some works try to find the cause of failures on distributed transaction environments with good times of response (Gao et al., 2004). One of them is about failure detection on heterogeneous environments as a NP-hard problem. Its approach is based on a dependency matrix of transactions versus resources and consider only binary dependencies i.e. a 0/1 matrix. Another work deals with the concept of self-regeneration introduced as a survival mechanism of systems that reduces the role of human experts. This work is focused on security and shows as an application the project CSISM, which implements multi-layer reasoning with fast reaction rules designed to take effective defensive actions within 250 ms after the initiation of the attack (Atighetchi and Pal, 2009).

Self-healing has been proposed for operating systems and distributed network environments (Rott, 2007). Rott decomposed this process in four main components: Monitoring, Adaptation, Interpretation and Resolution. By adopting the behavior of human administrators, also defined an optimal self-healing process in a computer environment into three stages: prevention, first aid and immunization. Rott considered as an example the ability to restore a service from an XML policy, which was implemented in Solaris 10. Some research demonstrated that it is possible to build self-healing operating systems through simple and effective techniques such as code reloading, component isolation and automatic restarts (David and Campbell, 2007).

A code injection mechanism for Java to introduce self-healing in object-oriented applications also has been proposed (Fuad et al., 2006). The model includes sensors that capture the state of the variables before calling the functions and encapsulating the exceptions. When any runtime failure occurs, the failure is notified and the system tries to reconstruct the unsuccessful method, so that it could be restarted at the point where the failure occurred. Otherwise, the system notifies the system administrator and some actions are executed like log generation. Fuad remarked that the code must be analyzed and the autonomic functionality should be

inserted in such a manner that it is separated from the service functionality of the legacy system. Also, a framework based on Java annotations was presented. This framework creates and builds applications with self-healing using a simple language of annotations (Breitgand et al., 2007).

Self-healing over networks was performed by injecting different types of faults to a network during training using cost-sensitive fault remediation (Littman et al., 2004). In cost-sensitive fault remediation, a decision maker is responsible for repairing a system when it breaks down. To narrow down the source of the fault, the decision maker can perform a test action, at some cost, and repair the fault if a repair action can be carried out.

A framework to specify, validate and generate autonomic systems, called Autonomic System Specification Language (ASSL) presented concrete results to specify a self-healing behavior model for NASA swarm-based exploration missions. The system send messages from a worker similar to heartbeats, or messages with a diagnosis (Vassev and Hinchey, 2009; Vassev and Paquet, 2007). A mechanism of self-healing for resource allocation using Ant Colony Optimization is also presented (Zhou et al., 2008). The obtained results are scalable to different kind of problems.

NASA has an initiative to carry out explorations in asteroid belts in a project called ANTS (Autonomic Nano Technology Swarm), based on autonomic computing (Truszkowski et al., 2004). The system has specialized workers to obtain information about asteroids, a central agent that gives a global goal and some messengers that send signals between the agents and the spatial station. NASA prototypes offer autonomic properties and these are defined in the architecture design that implements a wide level of autonomous and intelligent agents. These prototypes manage concepts, like specialization, in which an agent is designed to carry out a specific work and can redefine its task, it can also adapt itself to the environment and learn from its work or be easily replaced for other, if it has high-level failures. A concept mission is currently being planned to be launched between the years 2020 and 2030 as functional prototypes (Truszkowski et al., 2006).

In order to work self-healing from a generic perspective, taking some of the previous works as that of space exploration (ANTS) and the challenge of reproducing a system that captures part of the complexity of the real world, a virtual world in which termites that can carry out a task in a given environment is created. The termites simulator is an environment that includes the interaction among multiple elements, providing a more general solution instead of defining self-healing over operating systems or software, providing a motivation and a possible future extension not only for an application for self-healing in software but also for being extended to a hardware one.

Swarms have self-organization that makes them interesting. Considering that from self-organization it can be ob-

tained self-administration like an emergent property (Bicocchi and Zambonelli, 2007), not only it is possible to obtain self-administration but also self-healing. In this paper self-healing is studied from a perspective of artificial life that is based on the emergency and self-organization ideas, with many elements that interact with others through local rules and a synthetic approach would be adopted in which behaviors are understood throughout the construction of the same ones, using computer simulations (Langton, 1989).

Agents are called "termites" because they have social organization (only the worker termites are modeled). Feeding of termites is carried out for trophallaxis, it means that food is stored in their stomach and it is transferred among members of a community through mouth-to-mouth or anus-to-mouth feeding (Wikipedia, 2011). In this case also the pheromones define a communication mechanism.

In this paper, a simulator of a termite's swarm, a failure system, and a self-healing mechanism are introduced. Termites were modeled as agents with a virtual machine that execute instructions about motion and diagnosis. An Ant Colony System algorithm (ACS) was used to locate two points of food in the space. A failure was defined as a bad copy of a base program of a termite. Agents also diagnose others using language games and Q-learning.

Language games involves local interactions between two agents (a speaker and a hearer), in an environment with other agents, objects and situations. Some games allow the speaker to make the hearer perform an action (Steels and Vogt, 1997). The language game of diagnosis in this paper consists of one question of diagnosis about the programs of the termites; if the hearer termite does not have the code line that the speaker is expecting, the speaker rewards the diagnosis question. After recognizing the error, a voting parameter about failures is updated on each hearer termite, using a vector called belief vector of failures per each termite.

This paper is organized in the following way: first, the agents and the binary programs of the termites are described. The second part deals with the failure system, the diagnosis mechanism and the self-healing algorithm. Finally, the experiments with 100 and 200 termites are performed, with 10, 30, 50 and 70 percent of the sick termites at the beginning. Results are organized in terms of sick termites and termites that were healed.

## The Termites System

### Agents and ACS

Agents are termites that look for food, carry it, take it to the nest and continue searching for more food using Ant Colony System (Dorigo and Gambardella, 1997). The world is a toroidal space initialized with a pheromone value of zero for the termite nest and all termites start from this position with the simulation. Two points of food were defined with a pheromone value of one and the other world positions have a pheromone value of 0.5. Termites are represented as white

squares if they are looking for food and blue squares if they are carrying food and have eight possible movements: none, down, left, right, up, upleft, upright, downright and downleft (Fig 1).

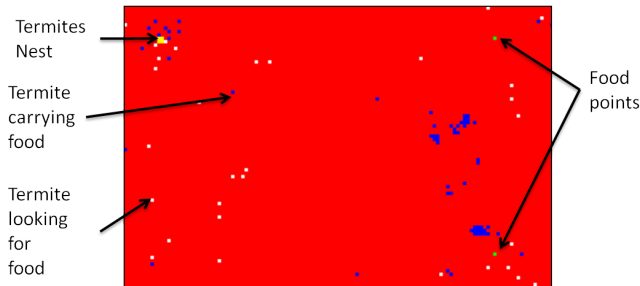


Figure 1: The Termites World

Termites start making random movements when looking for food and when they finally reach the food its color changes from white to blue and the pheromone production starts. Pheromone values in vicinity of termite and the search status (seeking, carrying) are the input of algorithm to select an action (Fig 2). If termite is looking for food, the first direction with the less amount of pheromone is chosen and if the termite is carrying food, the termite chooses the first direction with more pheromone. Then the termite moves in this direction, and the pheromone of termites and world pheromone are updated (Eq 1 and 2).

$$tph = (tph + 0.01 * (0.5 - tph)) \quad (1)$$

$$wph(x, y) = wph(x, y) + 0.01 * (tph - wph(x, y)) \quad (2)$$

Where:

- $tph$  is the pheromone of the termite.
- $wph(x, y)$  is the pheromone of world in the new location of the termite (x,y).

If the termite reaches its nest, its pheromone value is updated to 0, if the termite reaches a food point the pheromone of the termite gets a value of 1.

## The Termites Programs

Each agent has a simple program, which is executed line by line, that encapsulates the Ant Colony System algorithm and the mechanism of diagnosis based on language games. Each program is a vector of binary values that represent the sensations and actions to be performed by the termite. The base program of termites is exposed in Table .  $sSeek$  is a sensor that indicates if the termites are looking for food,  $sCarry$  indicates if the termites are carrying the food and  $sNeigh$  indicates if termites have only one neighbor. Neighbor and

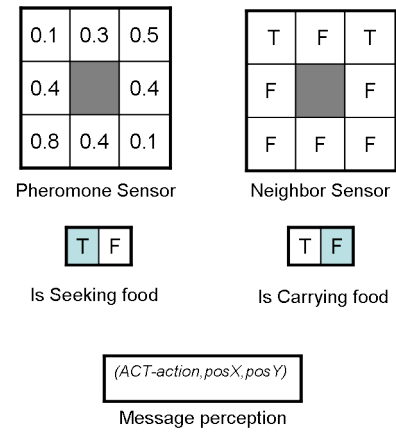


Figure 2: The Termites Sensors

pheromone sensors are defined in the Moore neighborhood  $r = 1$  with center in the termite location (Fig 2).  $acSeek$  and  $acCarry$  are simple instructions that execute the Ant Colony System algorithm and  $acDiag$  starts a diagnosis. The first instruction of the base program (Table ), is generated based on the rule: "if the termite is looking for food and the termite does not have one neighbor, then the termite has to look for food".

sSeek	sCarry	sNeigh	acSeek	acCarry	acDiag
1	0	0	1	0	0
0	1	1	0	0	1
0	1	1	0	1	0
0	1	0	0	1	0
1	0	1	0	0	1
1	0	1	1	0	0

Table 1: Base program for a termite

Each termite has an interpreter for its program. The interpreter takes each line of code and compares it with the perception of each sensor. If the line of code matches the perceptions, then the action indicated in the code line is performed. If more than one action is specified, the interpreter returns the action with the greatest priority. Priority is defined in the following order:  $acSeek > acCarry > acDiag$ .

## Failures definition

Program failures are simulated as bad coding from the beginning. Each termite has a variation of the program that the "queen" has (the base program). The programs are copied with a failure probability, it means not all termites will have programs with failures. For example, a failure probability of 0.1 means that approximately the 10% of the population have a failure.

A failure is a change in a random bit of the code per ter-

mite, so each termite has a different failure and it makes that the termites act in unexpected ways (Fig 3).

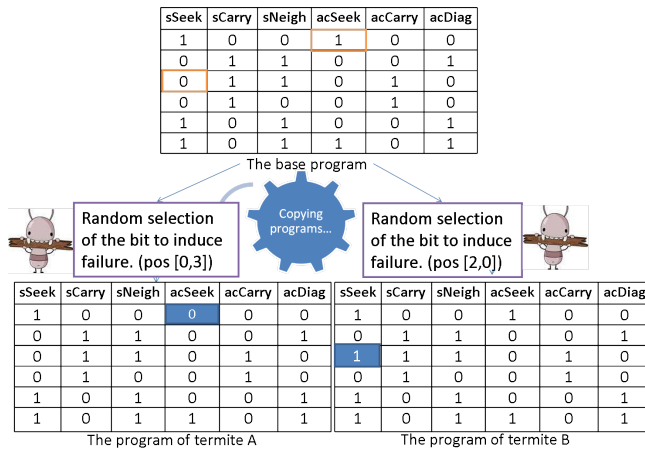


Figure 3: Failure selection for two termites

## Diagnosis Mechanism

### Diagnosis based on Language Games

A Language game is a sequence of local interactions between two agents (a speaker and a hearer) located in a specific environment (Steels, 2001). Some language games allow agents to identify objects in the environment using linguistic means and others allow the speaker to obtain actions from the hearer (Steels and Vogt, 1997). Some of language games were taken to design the mechanism of diagnosis for the programs.

A termite can send messages to a world location, if there is a termite in this place and the termites are neighbors. A diagnosis is started if a termite receives a message in its current position. This termite has to remain at this location, to clean the message from the world location and to reply the message. The diagnosis is encapsulated in the `acDiag` instruction in the termites program and was modified following the process below (Diagnose instructions are defined on Table 2):

- **Making contact:** Two agents are physically close (they are neighbors) and make contact with each another. One assumes the role of speaker and the other is the hearer.
- **Start Diagnosis:** The speaker chooses one line from its program (using Q-learning) and sends the codeline to the hearer location using the `RUNINSTR` instruction.
- **Action:** The hearer reviews its program, in this case compares its code with the line of code given by the speaker, and reports whether its program has this instruction or not (`INSTRRES` instruction). If the hearer does not have this line, the code line and a vote are added to a vector of possible code lines.

Instruction(syntax)	Definition
<code>RUNINSTR</code> ( <i>RUNINSTR-codeline, x, y</i> )	Indicates to the hearer a code line of the program from the speaker <i>code-line</i> and the position of the speaker ( <i>x, y</i> )
<code>INSTRRES</code> ( <i>INSTRRES-result, x, y</i> )	Indicates to the speaker if the hearer has the codeline or not and the current position of the hearer ( <i>x, y</i> )

Table 2: Diagnose instructions

- **Feedback:** If the hearer has this instruction, the diagnosis ends in failure (it does not discover a possible failure), the question of diagnosis about this line is punished using Q-learning. Otherwise, the hearer receives a positive vote for this code line, and the diagnose question is rewarded using Q-learning.

Q-learning (Watkins, 1989), is used to optimize the questions of diagnosis. There are questions of diagnosis about each code line per agent and weights associated with each code line which are stored in a vector of diagnosis questions.

If an error is detected (the hearer does not have the speaker's line), the question of diagnosis about this line of code receives a reward and otherwise the question receives a punishment. The goal of the agents in Q-learning is to maximize their total reward (Alpaydin, 2004). Questions of diagnosis with the greatest value are selected; if there is more than one question with the same greatest value, we choose the first one in the diagnosis vector.

The following equation is the reward when a failure is diagnosed:

$$d[c] = d[c] + \eta * (r + \gamma * \text{Max}_i(d[i]) - d[c]) \quad (3)$$

If a failure is not found, the following punishment for the question of diagnosis is given:

$$d[c] = d[c] - \eta * (r + \gamma * \text{Max}_i(d[i]) - d[c]) \quad (4)$$

Where:

- $d$  is the vector of weights about diagnosis questions.
- $c$  is the selected codeline for the diagnosis.
- $\eta$  is the learning rate ( $0 < \alpha < 1$ ).
- $r$  is the reward for taking the action.
- $\gamma$  is the discount factor for the maximum of the weights.

## Voting system

Each termite has a vector called belief vector of failures which stores the feedback of the diagnosis based on language games (a vote is added if the hearer does not have the code line that the speaker indicated). In this case, a value of 1 is added for this line of code if it belongs to the vector, otherwise the code line is added to this vector with a vote equal to 1. Table , shows four votes for the code line 100100, and three votes for the line 101011.

codeline	votes
100100	4
101011	3
101100	2
100101	1
110001	1

Table 3: belief vector of failures for a termite

## Self-healing

Self-healing is defined using some concepts of evolutionary algorithms. Evolutionary algorithms (EA) are optimization techniques based on the principles of natural evolution (Holland, 1992). First, a threshold was defined for the code lines in the belief vector of failures. If a code line of the belief vector of failures reaches this threshold (five votes in this case), it is introduced in a random position of the termite program, instead of adding another line. It could be considered like an operator of an EA. When this operator is applied, the introduced code line and its votes are removed from the belief about failures vector of this termite and the termite will disable the diagnosis instruction (Termite is sick so it cannot diagnose others), which is useful for avoiding failure propagation (Fig 4).

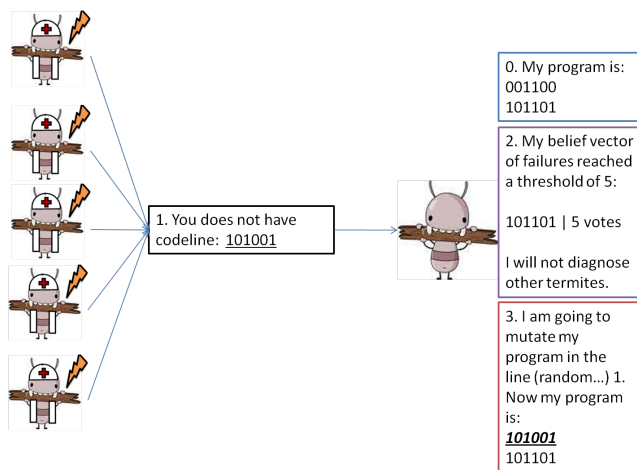


Figure 4: Self-healing process

## Dynamic of the process

Each termite gets their programs from the queen (base program). The base program is copied to all the termites and some termites of the population get bad copies of their programs (see failures definition section to get details). Some termites will be healthy and others will be sick and will act in unexpected ways. After that, the termites load and execute their programs. Thanks to the program, the termites know that they must look for food, carry food or make diagnostics.

Sick termites can diagnose healthy termites, so if a healthy termite receives bad diagnosis from sick termites (reach the threshold of the belief vector of failures), a code line of the healthy termite would be replaced and the healthy termite can get sick (see Self-healing section to get details) and disable its diagnosis instruction. In the same way a sick termite that is diagnosed by healthy termites, change their code, disable their diagnose instruction to avoid failure propagation and can be healed. With the time, the self-healing mechanism of programs avoid failure propagation and to induce changes in the lines of code of the sick termites decreasing disease.

## Experiments and Results

A virtual world was defined and each agent and food point were given a size of 1x1. For the Q-learning equations (Eqs 3 and 4) the following parameters were set:  $\eta = 0.01$ ,  $\gamma = 0.06$ , and a  $r = 1$ . Each question of diagnosis has an initial weight of  $1/\text{codelines}$ . There was a population of 100 and 200 termites with 0.1, 0.5 and 0.7 as the probability of failure (pf) in the program for the population at startup (see the faiures definition section for details). Code to validate if a program has been healed was introduced, but the agents have no knowledge about it.

Each experiment was performed 30 times, with 100000 iterations (movements per termite) per experiment. Data in Tables 4 and 5 presents the mean and the standard deviation of the experiments in terms of:

- PF = probability of Failure
- TS = Termites Sick at the Beginning are the termites that get sick by bad copy of their programs.
- TSBD = Termites Sick by Bad Diagnosis are the termites that get infected by bad diagnosis.
- TH = Termites Healed are termites which changed their code and got a code with the same instructions of the base program.
- TSDS = Termites Sick During Simulation are all the termites that got sick during the simulation (TS+TSBD).
- TSAS = Termites Sick After Simulation (TSDS - TH).



PF	TS	TSBD	TH	TSAS
0.1	9.76 ± 2.31	7.13 ± 4.59	16.4 ± 5.92	0.5 ± 0.68
0.3	30.2 ± 4.32	18.6 ± 5.44	42.43 ± 5.70	6.4 ± 4.07
0.5	49.6 ± 7.43	24.23 ± 6.60	47.06 ± 7.89	26.7 ± 6.14
0.7	68.5 ± 10.02	20.1 ± 5.89	20.2 ± 11.34	68.4 ± 13.92

Table 4: Experiments with 100 termites.

PF = probability of Failure, TS = Termites Sick at the Beginning, TSBD = Termites Sick by Bad Diagnosis, TH = Termites Healed, TSAS = Termites Sick After Simulation

PF	TS	TSBD	TH	TSAS
0.1	20.26 ± 4.64	9.63 ± 5.54	29.26 ± 7.89	0.63 ± 1.12
0.3	57.30 ± 7.42	27.53 ± 9.37	72.53 ± 8.67	12.30 ± 8.73
0.5	97.87 ± 15.24	53.43 ± 9.86	97.70 ± 15.59	53.60 ± 24.23
0.7	134.97 ± 9.52	32.27 ± 8.03	48.17 ± 13.52	119.07 ± 23.72

Table 5: Experiments with 200 termites.

PF = probability of Failure, TS = Termites Sick at the Beginning, TSBD = Termites Sick by Bad Diagnosis, TH = Termites Healed, TSAS = Termites Sick After Simulation

To determine if the algorithm is efficient, a t-test for related samples was performed with the following hypothesis. Results are organized in terms of total sick termites that got sick during the simulation (TSDS) and termites sick at the end of the simulation (TSAS):

- $H_o$  : the mean of the termites that got sick during the simulation TSDS ( $TSDS = TS + TSBD$ ) is equal to the mean of sick termites at the end of the simulation TSAS ( $TSAS = TSDS - TH$ ).
- $H_a$  : the mean of the total of termites sick > the mean of the termites sick at the end of the simulations ( $TSDS > TSAS$ ).

A value of  $\alpha = .05$  is selected for the tests (this value is the most used in social sciences), this means that five times out of a hundred a statistically significant difference between the means is found even if there was none.

For experiments with 100 termites, the means showed a difference between the termites sick during the simulation and the termites sick at the end of the simulation. The difference between the means is 15.467, the value of  $t$  is 14.096 for experiments with 0.1 as the failure probability. In the experiments with 100 and 0.3 of failure probability the difference between the means is 42.433, the value of  $t$  is 40.750. With 100 and 0.5 of failure probability the difference between the means is 47.067, the value of  $t$  is 32.651. For 0.7 we have a difference between means of 20.233 and a  $t$  value of 9.773. For 100 termites and  $pf = 0.1$ , Table 7 presents a sig value greater than .05 but the Paired Samples Test of table 8 reveal a statistically reliable difference between the means. The null hypothesis is rejected in all cases, so the

Pf		Mean	Std. Deviation.	Std. Error mean
0.1	TSDS	16.167	5.977	1.091
	TSAS	.70	.952	.174
0.3	TSDS	48.833	7.368	1.345
	TSAS	6.40	3.490	.637
0.5	TSDS	73.833	10.952	1.999
	TSAS	26.77	15.542	2.838
0.7	TSDS	88.633	7.513	1.372
	TSAS	68.40	13.922	2.542

Table 6: Paired Samples Statistics (100 termites, N = 30)

Pf	Correlation	Sig
0.1	.045	.812
0.3	.660	.000
0.5	.879	.000
0.7	.582	.001

Table 7: Paired Samples Correlations between TSDS and TSAS (100 termites, N = 30)

algorithm is efficient for the 100 termites and  $pf$  (0.1, 0.3, 0.5, 0.7) (Tables 6 and 8).

For experiments with 200 termites, the means also showed a difference between the total sick termites and the termites sick at the end of the simulation. For  $pf = 0.1$  the difference between the means is 29.266, the value of  $t$  is 20.314. In the experiments with  $pf = 0.3$  the difference between the means is 75.533, the value of  $t$  is 45.848. With  $pf = 0.5$  of failure probability the difference between the means is 97.700, the value of  $t$  is 34.325. For  $pf = 0.7$  we have a difference between means of 48.167 and a  $t$  value of 19.518. The null hypothesis is rejected in all cases, so the algorithm also is efficient for 200 termites and  $pf$  (0.1, 0.3, 0.5, 0.7) (Tables 9, 10 and ).

Pf		mean	Std. Deviation.	Std. Error mean
0.1	TSDS	29.9	8.442	1.541
	TSAS	.633	1.129	.206
0.3	TSDS	84.833	13.774	2.515
	TSAS	12.30	8.730	1.594
0.5	TSDS	151.300	17.542	3.203
	TSAS	53.60	24.234	4.424
0.7	TSDS	167.233	13.566	2.477
	TSAS	119.07	23.718	4.330

Table 9: Paired Samples Statistics (200 termites, N = 30)

## Conclusions and Future Work

A mechanism of programs self-healing based on language games, Q-learning and evolutionary computing was pre-

<b>TSDS- TSAS pf = 0.1</b>	<b>Paired Differences pf = 0.1</b>			<b>TSDS- TSAS pf = 0.3</b>	<b>Paired Differences pf = 0.3</b>		
	Mean	Std. Deviation	Std. Error Mean		Mean	Std. Deviation	Std. Error Mean
	15.467	6.010	1.097		42.433	5.704	1.041
	95% Confidence Interval for the difference	Lower	13.223		95% Confidence Interval for the difference	Lower	40.304
	t	Upper	17.711		t	Upper	44.563
			sig				sig
	14.096	29	.000		40.750	29	.000

<b>TSDS- TSAS pf = 0.5</b>	<b>Paired Differences pf=0.5</b>			<b>TSDS- TSAS pf = 0.7</b>	<b>Paired Differences pf = 0.7</b>		
	Mean	Std. Deviation	Std. Error Mean		Mean	Std. Deviation	Std. Error Mean
	47.067	7.896	1.442		20.233	11.340	2.070
	95% Confidence Interval for the difference	Lower	44.11841		95% Confidence Interval for the difference	Lower	15.999
	t	Upper	50.01493		t	Upper	24.46781
			sig				sig
	32.651	29	.000		9.773	29	.000

Table 8: Paired Samples Test (100 termites)

<b>TSDS- TSAS pf = 0.1</b>	<b>Paired Differences pf = 0.1</b>			<b>TSDS- TSAS pf = 0.3</b>	<b>Paired Differences pf = 0.3</b>		
	Mean	Std. Deviation	Std. Error Mean		Mean	Std. Deviation	Std. Error Mean
	29.266	7.891	1.441		72.533	8.665	1.582
	95% Confidence Interval for the difference	Lower	26.320		95% Confidence Interval for the difference	Lower	69.298
	t	Upper	32.213		t	Upper	75.769
			sig				sig
	20.314	29	.000		45.848	29	.000

<b>TSDS- TSAS pf = 0.5</b>	<b>Paired Differences pf=0.5</b>			<b>TSDS- TSAS pf = 0.7</b>	<b>Paired Differences pf = 0.7</b>		
	Mean	Std. Deviation	Std. Error Mean		Mean	Std. Deviation	Std. Error Mean
	97.700	15.589	2.846		48.167	13.516	2.468
	95% Confidence Interval for the difference	Lower	91.879		95% Confidence Interval for the difference	Lower	43.120
	t	Upper	103.521		t	Upper	53.214
			sig				sig
	34.325	29	.000		19.518	29	.000

Table 11: Paired Samples Test (200 termites)

<b>Pf</b>	<b>Correlation</b>	<b>Sig</b>
0.1	.539	.002
0.3	.794	.000
0.5	.767	.000
0.7	.876	.000

Table 10: Paired Samples Correlations between TSDS and TSAS (200 termites, N = 30)

sented. The system diagnoses and heals failures in an efficient way even with a 70% of the sick population. We observed that each termite is able to identify its own failures given the diagnosis of others.

Local interactions in the mechanism of diagnosis allow

the system to be specialized in the detection of more than a failure at the same time even if the failure is different per termite. By running the simulation, it was observed that some sick termites caused bad diagnosis, which induced failures in other termites. However, the rule that states that a termite cannot diagnose other if a failure is detected (votes threshold = five), makes that after some iterations, the termites stop propagating the failure and the population continues evolving their code until programs are recovered and the number of programs that were bad is reduced. In all cases the healing mutations stop after several iterations (Tables 4 and 5).

In all the experiments performed, the mean of the termites sick during the simulation (TSDS) is greater than the mean of the termites sick at the end of the simulation (TSAS) (Tables 6 and 9), so the null hypothesis ( $H_o$  : the mean of the

termites that got sick during the simulation is equal to the mean of sick termites at the end of the simulation) is rejected given the statistical analysis. With the time, the self-healing mechanism avoids failure propagation and induces changes in the lines of code of the sick termites obtaining less sick termites that the termites sick during the simulation. In this way, self-healing is an emergent property that arises from local interactions between termites (diagnosis based on language games).

As future work we are going to include some improvements like allowing the termite to locate the code line of the failure and perform diagnosis to others even if a failure is detected.

## Acknowledgments

This work was supported by the DIB (Dirección de investigación sede Bogotá- Universidad Nacional de Colombia) and its program "Apoyo de la DIB a tesis de investigación en posgrados - Segundo Corte" code 12153, and by the alife research group - National University of Colombia.

## References

- Alpaydin, E. (2004). *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.
- Arora, H., Raghu, T., Vinze, A., and Brittenham, P. (2006). Collaborative self-configuration and learning in autonomic computing systems: Applications to supply chain. pages 303–304.
- Atighetchi, M. and Pal, P. (2009). From auto-adaptive to survivable and self-regenerative systems successes, challenges, and future. pages 98–101.
- Bicocchi, N. and Zambonelli, F. (2007). Autonomic communication learns from nature. *Potentials, IEEE*, 26(6):42–46.
- Breitgand, D., Goldstein, M., Henis, E., Shehory, O., and Weinsberg, Y. (2007). Panacea towards a self-healing development framework. In *Integrated Network Management, 2007. IM '07. 10th IFIP/IEEE International Symposium on*, pages 169–178.
- David, F. M. and Campbell, R. H. (2007). Building a self-healing operating system. pages 3–10.
- De Wolf, T. and Holvoet, T. (2003). Towards autonomic computing: agent-based modelling, dynamical systems analysis, and decentralised control. pages 470–479.
- Dorigo, M. and Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*.
- Fuad, M., Deb, D., and Oudshoorn, M. (2006). Adding self-healing capabilities into legacy object oriented application. pages 51–51.
- Gao, J., Kar, G., and Kermani, P. (2004). Approaches to building self healing systems using dependency analysis. 1:119–132 Vol.1.
- Holland, J. H. (1992). *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA.
- Kephart, J. and Chess, D. (2003). The vision of autonomic computing. *Computer*, 36(1):41–50.
- Langton, C. G. (1989). *Artificial Life: proceedings of an interdisciplinary workshop on the Synthesis and Simulation of Living Systems*. Addison-Wesley.
- Littman, M. L., Ravi, N., Fenson, E., and Howard, R. (2004). Reinforcement learning for autonomic network repair. pages 284–285.
- Nami, M. and Bertels, K. (2007). A survey of autonomic computing systems. pages 26–26.
- Nami, M. R. and Sharifi, M. (2007). Autonomic computing: A new approach. pages 352–357.
- Rott, A. (2007). Self-healing in distributed network environments. 1:73–78.
- Steels, L. (2001). Language games for autonomous robots. *Intelligent Systems, IEEE*, 16(5):16 – 22.
- Steels, L. and Vogt, P. (1997). Grounding adaptive language games in robotic agents. In *Proceedings of the Fourth European Conference on Artificial Life*, pages 474–482. MIT Press.
- Truszkowski, W., Hinchey, M., Rash, J., and Rouff, C. (2006). Autonomous and autonomic systems: a paradigm for future space exploration missions. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 36(3):279–291.
- Truszkowski, W., Rash, J., Rouff, C., and Hinchey, M. (2004). Asteroid exploration with autonomic systems. pages 484–489.
- Vashev, E. and Hinchey, M. (2009). Assl specification and code generation of self-healing behavior for nasa swarm-based systems. In *Engineering of Autonomic and Autonomous Systems, 2009. EASe 2009. Sixth IEEE Conference and Workshops on*, pages 77–86.
- Vashev, E. and Paquet, J. (2007). Assl - autonomic system specification language. In *Software Engineering Workshop, 2007. SEW 2007. 31st IEEE*, pages 300–309.
- Watkins, C. (1989). *Learning from Delayed Rewards*. PhD thesis, University of Cambridge, England.
- Wikipedia (2011). Trophallaxis from wikipedia. <http://en.wikipedia.org/wiki/Trophallaxis>.
- Zhou, R., Wei, R., Chen, G., Yang, Z., Shen, H., Zhang, J., and Luo, M. (2008). Ant colony inspired self-healing for resource allocation in service-oriented environment considering resource breakdown. In *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT '08. IEEE/WIC/ACM International Conference on*, volume 1, pages 66–69.

# How to Measure Group Selection in Real-world Populations

Simon T. Powers<sup>1</sup>, Christopher Heys<sup>1</sup> and Richard A. Watson<sup>1</sup>

<sup>1</sup>Natural Systems Group, ECS, University of Southampton  
stp2@ecs.soton.ac.uk

## Abstract

Multilevel selection and the evolution of cooperation are fundamental to the formation of higher-level organisation and the evolution of biocomplexity, but such notions are controversial and poorly understood in natural populations. The theoretic principles of group selection are well developed in idealised models where a population is neatly divided into multiple semi-isolated sub-populations. But since such models can be explained by individual selection given the localised frequency-dependent effects involved, some argue that the group selection concepts offered are, even in the idealised case, redundant and that in natural conditions where groups are not well-defined that a group selection framework is entirely inapplicable. This does not necessarily mean, however, that a natural population is not subject to some interesting localised frequency-dependent effects – but how could we formally quantify this under realistic conditions? Here we focus on the presence of a Simpson's Paradox where, although the local proportion of cooperators decreases at all locations, the global proportion of cooperators increases. We illustrate this principle in a simple individual-based model of bacterial biofilm growth and discuss various complicating factors in moving from theory to practice of measuring group selection.

## Group selection in theory and practice

Some argue that the theoretic principles of group selection are well developed and crucial for understanding evolution in natural populations (Wilson and Wilson, 2007; Okasha, 2006). Indeed, many artificial life models seeking to explain the evolution of cooperation make either explicit or implicit reference to group-level selection (e.g., Scogings and Hawick 2008; Goldsby et al. 2009; Wu and Banzhaf 2009). The group selection position, however, suffers from at least two serious problems. The first is whether the phenomena involved, though undisputed, formally require group selection concepts. The second is whether the idealised conditions they assume are applicable in natural populations. We briefly overview the standard model of multilevel selection and discuss these limitations. Our aim is to devise a practical theoretical approach to assess whether something interesting is happening in a natural population with respect to the scale of selection. As a practical exemplar, we have in

mind the possibility of group selection occurring within natural bacterial biofilms. Biofilms are formed when bacteria attach to a surface and develop into dense aggregations, and they are in fact the most common mode of bacterial growth (compared to well-mixed planktonic populations). Bacteria living in biofilms are known to engage in many cooperative interactions, including the sharing of various 'public goods' such as extra-cellular enzymes. Biofilms also exhibit collective properties, such as anti-biotic resistance, that are significantly different from those of free-living bacteria (Ghanoun and O'Toole, 2004). Accordingly, they have potential to serve as an ideal model empirical system for studying the transition to multicellularity (Penn et al., 2008). However to do so, we need to be able to connect idealised models of multilevel selection (for example, where groups are discrete and non-overlapping) with real-world biological systems (where the "groups" may simply be local neighbourhoods with no discrete boundary). In this paper, we discuss the theoretic and practical issues involved in studying multilevel selection in biofilms and other natural populations. We illustrate our discussion with a simple individual-based model of bacterial growth, in which growth rate depends upon the local concentration of a 'public good' that is costly to produce. As such, this system might be expected to fit standard theory on the evolution of cooperation. However in our individual-based model, as in many real-world cases, the groups are not discrete and so it is not immediately obvious how, if at all, a multilevel selection framework can be useful. How, for example, can we measure the relative strengths of within- and between-group selection if the groups do not have discrete boundaries?

Despite this practical difficulty, theoretical and philosophical work suggests that multiple scales of selection should still be present in such systems (Wilson, 1980; Sober and Wilson, 1998; Nowak and May, 1992). Here, we illustrate the use of Simpson's Paradox (Simpson, 1951; Sober and Wilson, 1998) as a quantifiable indicator of a group-level selection effect. Crucially, we illustrate that this need not rely on a priori knowledge of the exact group structure, or even on the presence of discrete group boundaries. A Simp-

son's paradox occurs when, although the proportion of cooperators *decreases* in every locality, the global proportion of cooperators nevertheless *increases*. This can be measured in situ and does not require comparison with a well-mixed population, nor that we know the exact evolutionary game (fitness function) that individuals are engaged in. Then, by measuring the magnitude of the discrepancy between local and global proportions of cooperators over a range of local scales, we can identify the effective selective scale in a natural population. We also illustrate several further complicating factors that arise in moving from idealised theoretic models to more realistic biological scenarios.

### The idealised model of multilevel selection and its limitations

The idealised model of multilevel selection involves a population of individuals that is divided into discrete (equal-sized) sub-populations or demes (Wilson, 1980; Sober and Wilson, 1998), Fig. 1.

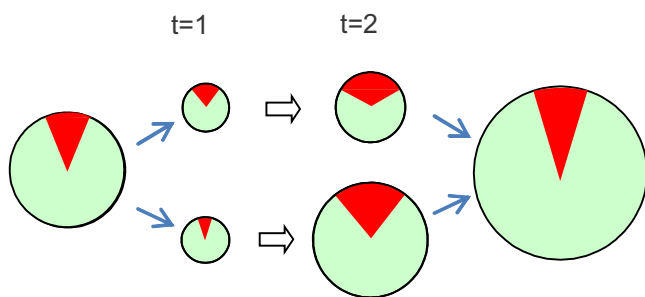


Figure 1: Growth of cooperators (green) & selfish individuals (red) living in groups. Individuals in each group (only two are depicted) are drawn randomly from a global population (left) such that the proportions of types (cooperators and defectors) varies slightly between groups. Groups with more cooperators grow more than groups with fewer cooperators and therefore contribute more individuals (specifically cooperators) to the global cell-count. Hence, the global proportion of cooperators increases (right).

Note this model assumes that localised fitness interactions are contained within neatly circumscribed groups. To sustain cooperation at high levels the population must be subject to multiple episodes of 'aggregation and dispersal', alternating between phases with a single 'migrant pool' (the global population or a representative sample thereof), and phases with multiple localised interaction groups. Without a group mixing stage, selfish behaviour would eventually go to fixation within each group founded by one or more selfish individuals (assuming Prisoner's Dilemma cooperative interactions; Powers et al. (2008); Powers (2010)).

**Is this really group selection?** It has been widely argued that this classic model shows nothing more than individual

selection given localised frequency-dependent effects (Maynard Smith, 1976; Nunney, 1985; Sterelny, 1996; Grafen, 1984), and hence does not involve group selection at all. That is, rather than saying *groups* with more cooperative individuals are fitter than *groups* with fewer cooperative individuals, we could equally say that *individuals* in groups with more cooperators are fitter than *individuals* in groups with fewer cooperators. In fact, our position is that if we could not explain the outcome of such models in terms of (context dependent) individual selection the result would be 'mystical' - that is, we would not have an evolutionary explanation at all. The behaviour of such models is fully explainable, as it must be, in terms of modified selective pressures on individuals given the group-living assumed. Nonetheless, it is at least interesting to note that the increase in levels of cooperation are consistent with the differential productivity of groups, i.e., more cooperative groups are fitter in terms of the genetic contribution they make to future generations, as well as consistent with the differential productivity of individuals, i.e. individuals in more cooperative groups are fitter in terms of the genetic contribution they make to future generations (Dugatkin and Reeve, 1994; Kerr and Godfrey-Smith, 2002). Indeed, this has to be the case because in this (very common) kind of multilevel selection model, group fitness is by definition the mean individual fitness of the group members (Damuth and Heisler, 1988; Okasha, 2006). However, even this pluralist position seems to be on shaky ground when the groups are not neatly defined. For example, how can we empirically measure group phenotypes (e.g., level of cooperation within the group) if we cannot identify discrete groups? In this case, a group based account will lose accuracy whereas the individual selection perspective remains undeniably precise (Godfrey-Smith, 2006).

**Is the standard model relevant to natural populations?** The standard model describes neatly partitioned sub-populations where the benefits of cooperative acts are distributed equally to members within each group, but not with members of other groups (Wilson, 1980; Godfrey-Smith, 2006). Such idealised conditions are likely to be rare in natural populations. Of course, the effect does not immediately vanish when groups are less neat. But in such cases, localised frequency-dependent selection seems a perfectly adequate explanation (Maynard Smith, 1976), and there seems to be little value in arguing for a 'group selection' account. Moreover, even if we wanted to retain a group selection framework, it is not clear how we could measure and quantify the differential productivity of groups in realistic scenarios where groups are somewhat ill defined (Godfrey-Smith, 2006).

These considerations should not lead one to conclude, however, that there is nothing of consequence presented in the idealised models (Okasha, 2006) nor that nothing interesting can happen in natural populations. But it is a bit tricky

to say what it is exactly, and more tricky to know how to measure it in a natural population. Certainly, if we were to assess the level of cooperation in a natural population, and then (assuming this were practically possible) assess it again in an artificially well-mixed version of the same experiment, we might see a difference in the two levels. This would at least tell us that localised frequency-dependent effects were significant in this system. But frankly, it does not sound all that interesting – it is rather obvious that selective pressures will be different in well-mixed populations if locally dispersed resources or public goods are involved. Simply examining the global frequency of cooperation tells us nothing about the mechanism behind its evolution, e.g., is cooperation a simple mutualism or is it individually-costly?

Moreover, although a comparison of well-mixed versus spatial or viscous populations is possible in synthetic simulations, the practicalities of say, mechanically mixing a biofilm or adding surfactants to break-up the extra-cellular matrix that holds cells together would not merely alter spatial relationships, but potentially affect many important environmental factors that could confound the result. We are left, therefore, with a significant gap between the theoretic idealisations of group selection and methodology that would be useful in practical situations (West et al., 2008).

An alternative is to look for a Simpson's paradox in situ. A Simpson's paradox clearly emphasises the crucial mechanics of multilevel selection (Sober and Wilson, 1998), see below, and it can be measured in situ so that it does not require disruption of the natural population structure.

### Group selection and Simpson's paradox

Simpson's paradox is a statistical phenomenon that arises when correlations or trends within sub-groups of a data set fail to represent the overall correlation when all the data is assessed together (Simpson, 1951; Sober and Wilson, 1998). Table 1 shows a very simple hypothetical example based on a group selection scenario. It shows the numbers of cooperators and selfish individuals in two groups, A and B, at two time points,  $t = 1$  and  $t = 2$ . Note that both groups show a *decrease* in the proportion of cooperators in this time interval, yet overall, from the same data, there is nonetheless an *increase* in the total proportion of cooperators.

It may be useful to clarify that at a given point in time, the average within-group proportion of cooperators can be different from the global proportion of cooperators. This is simply because the average within-group proportion weights all groups equally, whereas the global proportion is implicitly the same summation but with each group contribution 'weighted' in proportion to its size. In the example, at  $t = 1$  the groups are equal sized and the average within-group proportion and the global proportion are therefore the same. But in the second time point, the groups are different sizes and the average within-group proportion  $((31\% + 62\%)/2 = 46.5\%)$  is not equal to the global pro-

portion (51%).

In this example then, the growth trend paradox (i.e., cooperation decreases within groups but increases globally) is caused by the fact that one group grows much more than the other. Specifically, the B group, with twice the initial proportion of cooperators, is assumed to grow at about twice the rate as the A group in this example. So, although selfish individuals always grow faster than the cooperators in any given environment, some cooperators grow faster than some selfish individuals (specifically, when cooperators are in an environment of many other cooperators). Accordingly, because highly cooperative groups grow more, cooperators can increase in total proportion even though they decrease in proportion within each group.

### Using Simpson's paradox to indicate group selection

Simpson's paradox as a basis for group selection is well understood. However, it is generally not used as a direct indicator of group selection. Instead, the norm is simply to assess the global level of cooperation and see if it increases. But in practical experiments this is insufficient to conclude that group selection is responsible for such an increase. When the exact form of the evolutionary game that individuals are engaged in is unknown, due to numerous modes of interaction and multiple 'public goods' for example, or competition for multiple resources, it can be difficult to genuinely ascertain whether the 'cooperator' is really cooperating and whether the 'selfish' type is really selfish. That is, should we be surprised that the global level of cooperation increases, or is it a simple case of mutualism? The obvious control is to compare with a well-mixed population or to increase the diffusion rate in a spatial model, but aside from the practical difficulties of this in natural populations (even bacterial ones), this cannot maintain the 'all other things being equal' condition necessary to determine that only the localisation of interactions is producing the difference in results. Instead, by looking for a divergence between the average within-group and global proportions of types, we can both verify that the types are behaving as expected (that in any given environment the selfish individuals have the advantage) and identify a group selection effect if there is one. Thus Simpson's Paradox provides an in situ measurement of group selection in the sense that we do not need to disrupt groups to provide a control, and can therefore assess the effect that groups are having merely by observing how the frequencies of types change in the natural population.

To measure Simpson's Paradox in scenarios that have poorly defined groups requires an additional small step. For this we propose the following practical methodology for a spatially distributed population. Rather than attempt to define boundaries around one group and distinguish it from another, we can simply divide the physical space into equal-sized local regions and measure both the average local pro-

	$t = 1$			$t = 2$		
	Coop	Selfish	%Coop	Coop	Selfish	%Coop
A	2	4	<b>33%</b>	4	9	31%
B	4	2	<b>66%</b>	16	10	62%
Total	6	6	50%	20	19	<b>51%</b>

Table 1: Numbers of cooperative and selfish individuals in two hypothetical groups, illustrating Simpson’s paradox. Bold highlighting indicates the time point where the proportion of cooperators is highest. Note that within both group A and group B the proportion of cooperators decreases over this period, but overall, the proportion of cooperators increases.

portion of cooperators within all regions, and the global proportion of cooperators. If the selfish individuals are indeed selfish individuals then the average local proportion of cooperators must be always declining. But if, at the same time, the global proportion of cooperators is increasing then there is significant group selection activity.

Note that if every region exhibited approximately the same amount of total cell growth, then a paradox could not occur; but if some local regions are growing much faster than others (because local frequency-dependent fitness effects are sufficiently strong) a Simpson’s Paradox may be observed. In principle, it does not matter whether the space is divided into contiguous tiles (as we employ below), or whether regions are selected at random with random centres. But it does matter that regions are not selected in any manner that is biased by cell density, for that would amount to taking a weighted average. Taking a weighted average would necessarily make the local average the same as the global, and so would result in the local group dynamics disappearing from the analysis. This is the “averaging fallacy” described by Sober and Wilson (1998), which causes the appearance of group selection to vanish. For example, measuring the proportion of cooperators in the vicinity of each and every cell or within its radius of influence will bias measurements of local proportions in such a manner that dense areas contribute more to the average in exact proportion to how dense they are – in this case, the average local proportion cannot be different from the global proportion.

In the remainder of this paper we develop a simple individual-based model of bacterial growth, such as would apply to a locally-dispersing ‘public good’, to illustrate the use of this methodology and as a basis for discussion of several additional complicating factors that are important in its application. Of particular interest is the possibility of measuring the local proportions at several different spatial scales to determine the effective scale of selection.

## An individual-based model

### Bacterial Biofilms

In developing the following model we have bacterial biofilms in mind. Social evolution in bacterial systems is currently receiving considerable attention both as a model system of social evolution and because of the practical im-

plications of biofilms (Crespi, 2001; Griffin et al., 2004; Burmolle et al., 2006). Biofilms show a physical structure especially suited for localised fitness interactions via the formation of semi-isolated micro-colony structures (Hall-Stoodley et al., 2004). However, the following model is general – not dependent on any of the particulars that pertain to specific bacterial strains or types of fitness interaction. The vital assumptions are that there are two types of individual, that the presence of one of these types (but not the other) is beneficial to other individuals within a certain spatial radius, and that this type bears a cost for providing this benefit. For example, one type may be a wild-type strain of *Pseudomonas Aurigenosa*, that releases into the environment an enzyme useful for binding iron (Griffin et al., 2004). This enzyme can be understood as a ‘public good’ because it can be used by others within the diffusion radius of the molecule. The other type may be a selfish mutant strain that does not produce the public good and is therefore not burdened by its production, but can, like any other individual, benefit from the public good produced by cooperators.

### Model definition

The state of the model at any point in time is defined by a population of individuals each of which has a type (cooperate/selfish), an age, a location in continuous 2D space and a ‘reproductive potential’. Reproductive potential can be thought of as the resources the individual has accumulated over time. There is no explicit modelling of the public good, diffusion constants, extra-cellular matrix, or such like – and in the default model, cells do not move. At every point in time, the fitness potential of each cell is incremented by a fitness benefit,  $W$ . This is a function of both the individual’s own type, and of the number of cooperators in the local vicinity. Specifically, the fitness benefit of an individual is:

$$W = m + Pb - c, \quad (1)$$

where  $m = 1.5$  is a constant representing the intrinsic growth rate,  $P$  is the proportion of cooperators within a given radius,  $r_1 = 15$ , of the individual (including itself),  $b = 4$  is a constant representing the fitness benefit received from cooperators, and  $c = 0$  for selfish individuals and  $c = 1.8$  for cooperators is the cost of being a cooperator (i.e., the cost of producing the public good). This fitness



function is standard in evolutionary models of altruism (Wilson, 1980), and amounts to an  $n$ -player public goods game / Prisoner's Dilemma (Fletcher and Zwick, 2007).

The model proceeds by updating each individual, in each time step, according to Algorithm 1.

---

**Algorithm 1** Individual update algorithm.

---

1. The age is incremented by 1.
  2. If the age is 5 the cell dies.
  3. Otherwise, the fitness benefit is calculated (as above) and added to the individual's current reproductive potential.
  4. Whilst the reproductive potential  $> 4$ ,
    - (a) Reproduce, placing descendant cell in a new location according to a placement algorithm (see text). An offspring is an exact genetic clone of its parent.
    - (b) Decrement reproductive potential by 4.
- 

The model is initialised with equal numbers of cooperators and selfish individuals distributed uniformly at random. Each initial cell (and new cell from reproduction) is initialised with reproductive potential=0, and age=0. The placement algorithm may take account of competition for space (and possibly fail to produce an offspring if space does not allow) but by default it simply places an individual in a random location within a radius,  $r_2 = 5$ . Thus, an offspring is placed close to its parent.

Measuring the global proportion of cooperators is trivial. To measure the average local proportion of cooperators, the space is divided into contiguous square regions of size,  $r_3 = 15$  (note that the area of each square local region,  $(r_3)^2 = 225$ , in which local proportions are measured, is the same order of magnitude as the circular area over which a cooperator may affect other individuals,  $\pi(r_1)^2 = 707$ . See Fig. 5.).

In an advanced version of the model, cells are motile and move toward cooperators. This represents attraction towards concentration of the public good, for example. At each time step, a vector is calculated which is a distance-discounted sum of vectors to all other local regions, weighted by the number of cooperators in that region. The regions used are the same as those used for calculating the average local proportion of types. Each cell then moves a random distance  $d$  in the direction of this vector;  $d$  is uniformly distributed in the range 0 to  $15r_4$ , where  $r_4$  is a constant controlling the amount of movement.

### Model illustrations

We initialised each simulation with 150 cooperators and 150 selfish individuals, distributed randomly across a square grid of size  $250 * 250$ . Each simulation was repeated 50 times,

and the mean of both the average local and global proportions of cooperators recorded.

Figure 2 shows that although the initial distribution of bacterial cells is random, the cells grow into spatial clusters due to non-motility and the fact that offspring are placed close to their parents (as per *Model definition*).

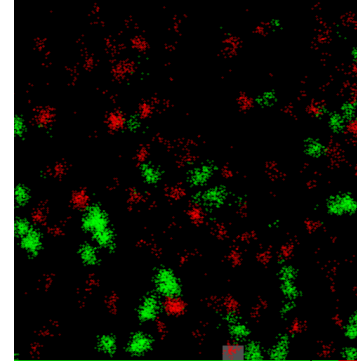


Figure 2: Illustration of biofilm growth in the model. Green cells are cooperators, red are selfish cheats.

From standard social evolution theory, we would not expect cooperation to increase or be stable in the absence of localised interactions (Wilson, 1980). Thus, in such cases we should not see a Simpson's Paradox, since without localised interactions there should be no difference in the growth-rates of different localities, *ceteris paribus*. We verified that this was the case in our model by making the radius of social interactions,  $r_1$ , equal to the size of the whole grid. Thus, each individual would experience the global proportion of cooperation for the purposes of determining their fitness. This corresponds to complete mixing of the public good, but not of the individuals themselves. Thus, we still measured the local proportion of cooperation across squares of size  $r_3 = 15$ . As Figure 3a shows, the global frequency of cooperation steadily declines in this case, and there is no observation of a Simpson's Paradox. This is because although there are still spatial groups in the system, membership of these groups does not affect fitness when the public good is global, and hence they are meaningless to evolution. This serves as an illustration of the fact that the groups we can readily observe in a system (e.g., the clusters in our model) may not be the same scale as the groups that matter for the evolution of cooperation (in the case of well-mixed public goods, the 'group' is the whole population).

On the other hand, in Figure 3b we set the radius of the public good to  $r_1 = 15$ . This represents localised interactions, and so we might expect cooperation to evolve. Moreover, we set the window size over which we measure local proportions of cooperators to be of this same scale ( $r_3 = 15$ ). In this case cooperation evolves, and we observe a difference between average local and global proportions of cooperation, and hence a Simpson's Paradox. It should

be noted that Simpson's Paradox is present even when the global proportion of cooperators is falling, so long as the average local proportion of cooperators is falling at a faster rate (e.g., generations 1-6 in Figure 3b). In this case there is a non-zero between-group component of selection, but this is weaker than within-group selection.

Figure 3b also illustrates that the paradox cannot be sustained indefinitely. This is because selfish individuals are fitter than cooperators sharing the same public good (same  $P$  value but  $c = 0$  in Equation 1). Thus, they must necessarily increase in frequency within each locality. As this happens,

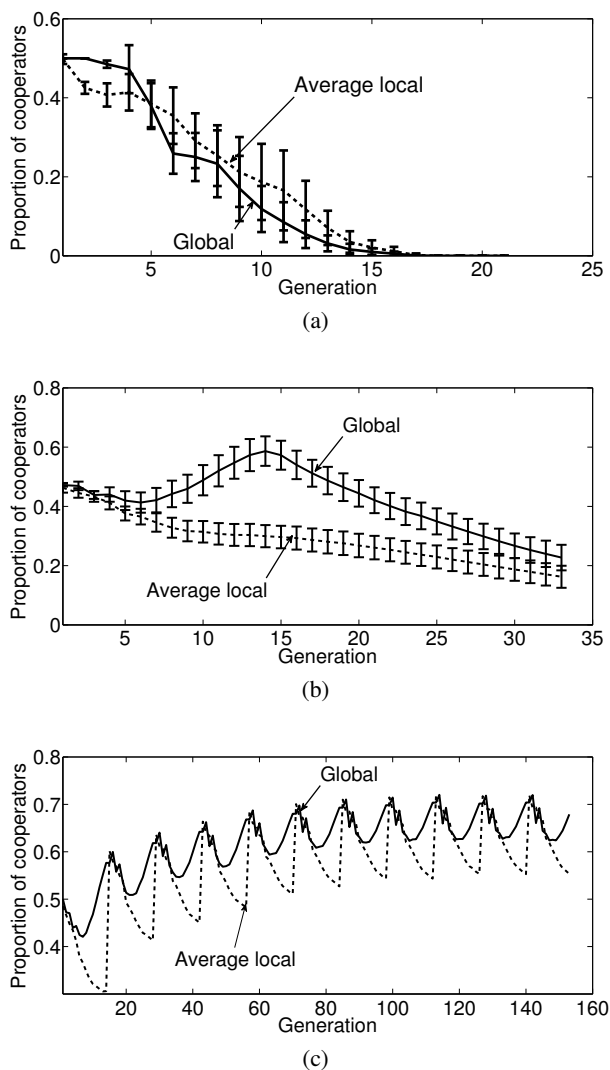


Figure 3: A) When the average of interaction,  $r_1$  covers the entire space, cooperation does not evolve and Simpson's Paradox is not observed. B) When  $r_1 = 15$  cooperation evolves, and there is a difference between local and global proportions. C) Multiple aggregation and dispersal cycles with  $r = 15$ .

the differential growth of different localities decreases, and hence the paradox reduces. In Figure 3b, the paradox peaks at 14 generations, after which the global frequency of cooperation starts to fall back down. This seemingly inevitable decrease in cooperation as the generations go by need not occur, however, if individuals are periodically mixed and redistributed in space (Sober and Wilson, 1998). Essentially this is because such a redistribution of individuals reestablishes variance in the proportion of cooperators (and hence in the amount of the public good) between groups, and so once again allows for differential group productivity to have an effect and create a paradox. This is illustrated in Figure 3c, where dispersal from clusters and global mixing occurs every 14 generations. These dispersal events explain the see-saw shape of the average local curve: at each dispersal event, the average local proportion is returned to the global proportion of cooperators. Dispersal is known to occur in natural biofilms (Ghannoum and O'Toole, 2004) (although simultaneous and complete mixing is a simplifying assumption of our model), and the single-celled bottleneck in the development of multicellular organisms provides a similar redistribution of genetic variance (Maynard Smith and Szathmáry, 1995; Michod, 1999). Thus, some degree of dispersal is likely to be important in maintaining cooperation in natural populations (West et al., 2002), and may actually be an evolutionary adaptation at least partly for this purpose (Maynard Smith and Szathmáry, 1995; Michod, 1999).

Figure 4 shows the effect of cell motility on the observation of Simpson's Paradox. Again, from standard theory we would expect increasing motility to reduce global levels of cooperation. We see that increasing motility decreases Simpson's Paradox. This is because it increases the heterogeneity of localities, making their  $P$  values more similar and hence the differential in group productivity lower.

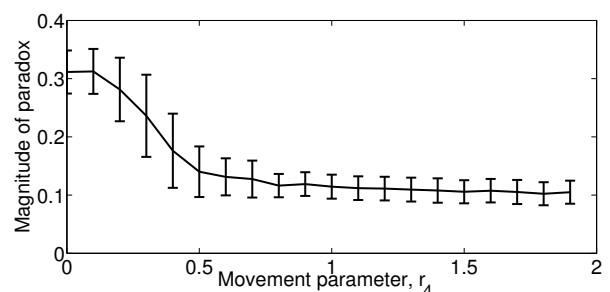


Figure 4: Effect of increasing cell motility on the observation of Simpson's Paradox. Error bars show standard deviation.

Figure 5 shows how the peak observation of a Simpson's Paradox changes depending on the scale at which local proportions of cooperators are measured. Observation of the paradox will peak when this scale corresponds to the actual scale of social interactions in the system, e.g., to the radius

in which the public good is shared. The peak in Figure 5 is where the measured locality size corresponds, approximately, to  $r_1$ , the actual scale of interaction. Measuring Simpson's Paradox using different local scales could thus be used to determine the actual scale of social interactions in a real-world system, where this may well not be known *a priori*.

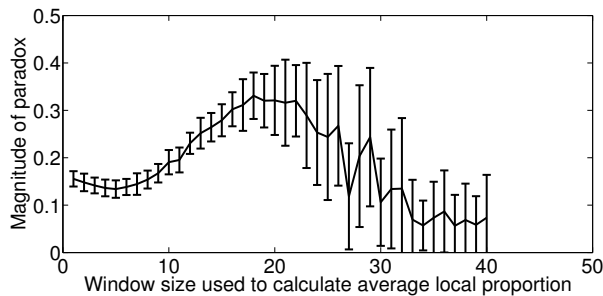


Figure 5: Effect of the magnitude of the locality size measured on the observation of Simpson's Paradox (difference between local average and global proportion of cooperators). The observed paradox is strongest when the measured locality size corresponds to the actual scale of social interaction; measurements were taken after the number of generations that yielded the peak difference between global and local frequencies, for each window size. Error bars show standard deviation. The length of the error bars increases with the window size because a larger window size corresponds to fewer localities and hence fewer samples to average over.

## Discussion

We have presented a methodology for measuring the effect of group-level selection in natural populations. Real-world populations may often not be formed of clearly observable groups with discrete boundaries, which makes the application of standard multilevel selection theory non-trivial. In particular, theoretical techniques for measuring the strength of group selection, such as the Price Equation or contextual analysis, rely on being able to measure properties of discrete groups (Godfrey-Smith, 2006). Thus, their application to systems such as bacterial biofilms remains problematic.

Here, we have suggested observation of Simpson's Paradox as a way to quantify the effect of group-level selection in a natural population. It is now widely appreciated that Simpson's Paradox, the difference between average local and global frequencies of cooperation, will be present whenever individually-costly cooperative behaviours evolve (Sober and Wilson, 1998). Moreover, its presence indicates multiple scales of selection in a system (Sober and Wilson, 1998). However, discussions of Simpson's Paradox have so far remained in the theoretical domain. In particular, illustrations of it have, to our knowledge, only been conducted in models with discrete group boundaries. By contrast, we

have shown that Simpson's Paradox can be readily measured in populations where individuals are continuously distributed throughout space. Thus, the exact group structure does not have to be known *a priori* for this technique to be applied. We have illustrated the measurement of Simpson's Paradox in such a case with an individual-based model of public goods production in bacterial biofilms.

Significantly, measurement of Simpson's Paradox can be used to determine the effective group structure in a natural population. Specifically, the difference between average local and global proportions of cooperation will peak when the size of localities measured is of the same scale as that over which the public good is shared. That is, when the measurement window size matches the scale of fitness-affecting social interactions. Wilson (1980) terms the scale over which social interactions occur "trait groups". He stresses that the groups which matter to natural selection are subsets of individuals in which fitness-affecting interactions occur, and that these subsets may not correspond to the apparent groups that are most readily observable in a population. For example, although discrete clusters may be observable in a biofilm, these may not correspond to the radius over which a public good diffuses. Varying the window size over which the change in local proportions of cooperators is measured, and looking for the peak difference with the global proportion, can identify the effective trait groups in the population. Searching for the trait groups in this way can be done by image analysis at the end of the experiment – the experiment does not have to be re-run in order to measure Simpson's Paradox on different scales. Regarding biofilms, one may also measure local proportions using regions that specifically enclose micro-colonies to see if micro-colony structure is a stronger selective unit than arbitrary local regions. That is, our methodology can be used to determine whether the micro-colonies correspond to trait groups, or whether the trait groups are in fact smaller or larger.

In future work, it would be interesting to investigate whether the Price Equation can be meaningfully applied to the appropriate window size. In particular, our methodology identifies non-arbitrary groups. Thus, once we have identified the effective trait group size, we could calculate the covariance between group character (local proportion of cooperators), and group productivity. Likewise, we could calculate the covariance between individual character (cooperator or not) and individual fitness (number of cell divisions). Our methodology also fits within a kin selection framework (Hamilton, 1964), as used by Griffin et al. (2004) to study bacterial social evolution, for example. Finding the trait groups corresponds to finding the scale at which genetic relatedness should be measured in a natural population.

**Acknowledgements** Thanks to Alex Penn, Jeremy Webb and Lex Kraaijeveld.

## References

- Burmolle, M., Webb, J. S., Rao, D., Hansen, L. H., Sorensen, Soren, J., and Kjelleberg, S. (2006). Enhanced biofilm formation and increased resistance to antimicrobial agents and bacterial invasion are caused by synergistic interactions in multispecies biofilms. *Appl. Environ. Microbiol.*, 72(6):3916–3923.
- Crespi, B. J. (2001). The evolution of social behaviour in microorganisms. *Trends Ecol. Evol.*, 16(4):178–183.
- Damuth, J. and Heisler, I. L. (1988). Alternative formulations of multilevel selection. *Biol. Philos.*, 3(4):407–430.
- Dugatkin, L. and Reeve, H. (1994). Behavioral ecology and levels of selection: Dissolving the group selection controversy. *Advances in the Study of Behavior*, 23:101–133.
- Fletcher, J. A. and Zwick, M. (2007). The evolution of altruism: Game theory in multilevel selection and inclusive fitness. *J. Theor. Biol.*, 245(1):26–36.
- Ghannoum, M. A. and O'Toole, G. A., editors (2004). *Microbial Biofilms*. ASM Press, Washington, DC.
- Godfrey-Smith, P. (2006). Local interaction, multilevel selection, and evolutionary transitions. *Biological Theory*, 1(4):372–380.
- Goldsby, H., Knoester, D., Clune, J., McKinley, P., and Ofria, C. (2009). The evolution of division of labor. In *Proceedings of the European Conference on Artificial Life (ECAL)*.
- Grafen, A. (1984). Natural selection, kin selection and group selection. In Krebs, J. and Davies, N., editors, *Behavioural Ecology*, chapter 3, pages 62–84. Blackwell Scientific Publications, Oxford, 2nd edition.
- Griffin, A. S., West, S. A., and Buckling, A. (2004). Cooperation and competition in pathogenic bacteria. *Nature*, 430:1024–1027.
- Hall-Stoodley, L., Costerton, J. W., and Stoodley, P. (2004). Bacterial biofilms: from the natural environment to infectious diseases. *Nat. Rev. Microbiol.*, 2(2):95–108.
- Hamilton, W. D. (1964). The genetical evolution of social behaviour. I. *J. Theor. Biol.*, 7(1):1–16.
- Kerr, B. and Godfrey-Smith, P. (2002). Individualist and multilevel perspectives on selection in structured populations. *Biol. Philos.*, 17:477–517.
- Maynard Smith, J. (1976). Group selection. *Q. Rev. Biol.*, 51:277–283.
- Maynard Smith, J. and Szathmáry, E. (1995). *Major Transitions in Evolution*. W. H. Freeman/Spektrum, Oxford.
- Michod, R. E. (1999). *Darwinian Dynamics: Evolutionary Transitions in Fitness and Individuality*. Princeton University Press.
- Nowak, M. A. and May, R. M. (1992). Evolutionary games and spatial chaos. *Nature*, 359:826–829.
- Nunney, L. (1985). Group selection, altruism, and structured-deme models. *Am. Nat.*, 126(2):212–230.
- Okasha, S. (2006). *Evolution and the Levels of Selection*. Clarendon Press.
- Penn, A. S., Watson, R. A., Powers, S. T., Webb, J. S., Kraaijeveld, A. R., Conibear, T. C., and Bigg, Z. (2008). Mechanisms for the initiation of multicellularity in bacterial biofilms. In *Proceedings of the 10th International Conference on the Simulation and Synthesis of Living Systems*, page 794.
- Powers, S. T. (2010). *Social Niche Construction: Evolutionary Explanations for Cooperative Group Formation*. PhD thesis, University of Southampton.
- Powers, S. T., Penn, A. S., and Watson, R. A. (2008). The efficacy of group selection is increased by coexistence dynamics within groups. In *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 498–505, Cambridge, MA. MIT Press.
- Scogings, C. and Hawick, K. (2008). Altruism amongst spatial predator-prey animats. In *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 537–544, Cambridge, MA. MIT Press.
- Simpson, E. H. (1951). The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society B*, 13:238–241.
- Sober, E. and Wilson, D. S. (1998). *Unto Others: The Evolution and Psychology of Unselfish Behavior*. Harvard University Press, Cambridge, MA.
- Sterelny, K. (1996). The return of the group. *Philos. Sci.*, 63(4):562–584.
- West, S. A., Griffin, A. S., and Gardner, A. (2008). Social semantics: how useful has group selection been? *J. Evol. Biol.*, 21(1):374–385.
- West, S. A., Pen, I., and Griffin, A. S. (2002). Cooperation and competition between relatives. *Science*, 296(5565):72–75.
- Wilson, D. S. (1980). *The Natural Selection of Populations and Communities*. Benjamin/Cummings, California.
- Wilson, D. S. and Wilson, E. O. (2007). Rethinking the theoretical foundation of sociobiology. *Q. Rev. Biol.*, 82(4):327–348.
- Wu, S. X. and Banzhaf, W. (2009). Investigations of Wilson's and Traulsen group selection models in evolutionary computation. In *Proceedings of the European Conference on Artificial Life (ECAL)*.

# Using statistical inference for designing termination conditions ensuring convergence of Evolutionary Algorithms

David Roche<sup>1</sup>, Debora Gil<sup>2</sup> and Jesus Giraldo<sup>1</sup>

<sup>1</sup>Laboratory of Systems Pharmacology and Bioinformatics, Universitat Autònoma de Barcelona, Bellaterra, Spain

<sup>2</sup> Computer Science Department and Computer Vision Center (CVC), Universitat Autònoma de Barcelona, Bellaterra, Spain  
{david.roche,jesus.giraldo}@uab.es, debora@cvc.uab.es

## Abstract

A main challenge in Evolutionary Algorithms (EAs) is determining a termination condition ensuring stabilization close to the optimum in real-world applications. Although for known test functions distribution-based quantities are good candidates (as far as suitable parameters are used), in real-world problems an open question still remains unsolved. How can we estimate an upper-bound for the termination condition value ensuring a given accuracy for the (unknown) EA solution?

We claim that the termination problem would be fully solved if we defined a quantity (depending only on the EA output) behaving like the solution accuracy. The open question would be, then, satisfactorily answered if we had a model relating both quantities, since accuracy could be predicted from the alternative quantity. We present a statistical inference framework addressing two topics: checking the correlation between the two quantities and defining a regression model for predicting (at a given confidence level) accuracy values from the EA output.

## Introduction

Evolutionary Algorithms (EAs) are a class of stochastic optimization methods that simulate the process of natural evolution. EAs maintain a population of possible solutions that evolve according to rules of selection and other operators, such as recombination and mutation. Several evolutionary methodologies have been proposed for solving real world optimization problems: genetic algorithms (Holland, 1975), evolutionary strategies (Schwefel, 1995) and differential evolution (Storn and Price, 1997) among others. By their ability to optimize non-analytic multi-modal functions, EAs have been successfully applied to a wide range of real life problems, such as parameter estimation (Ravikumar and Panigrahi, 2010), pattern and text recognition (Rizki et al., 2002) and image processing (Cagnoni et al., 2008).

As any iterative technique, EA requires a stop criterion. Unlike optimization methods adapting a single initial value (which rely on real analysis theory), by their stochastic nature, there is not a solid mathematical theory ensuring con-

vergence of evolutionary methodologies in general (Safe et al., 2004; Bäck et al., 1997).

The simplest (and most extended (Safe et al., 2004; Price et al., 2005; Tagetiren and Suganthan, 2006)) stopping criterion consists in reaching a number of iterations or function evaluations. This stopping criterion is not useful by itself (the number of iterations that guarantee convergence significantly varies across problems (Safe et al., 2004)), though it can be necessary when used in addition with alternative criteria to ensure that the algorithm stops (Zielinski and Laur, 2008).

Existing approaches defining general alternative termination conditions address two issues. First, the definition of a quantity reflecting the amount of change between consecutive iterations and, second, the condition that such quantity should fulfill. The quantities reported in the literature (Zielinski and Laur, 2008; Safe et al., 2004) measure either the rate of change in the objective function (improvement-based) or the distribution of the evolving population (distribution-based). Concerning the termination condition, two different conditions are considered. The first condition terminates EA if the measure of the amount of change is below a given threshold. The second one terminates EA in the case that such measure is below a threshold for a number of generations. Improvement-based criteria may lead to early termination (possibly far from the optimum) due to the stochastic nature of EA (Zielinski and Laur, 2008). Meanwhile, distribution-based quantities compare to the accuracy of the solution (distance to the optimum) in terms of number of function executions, as far as suitable parameters (threshold and number of generations) for the termination condition are set (Zielinski and Laur, 2008).

A main limitation for application to real-world problems is that the parameters of the termination condition strongly depend on the function shape of the objective function (Zielinski and Laur, 2008). Another concern is that current approaches constrain to statistically comparing the number of iterations reached by the termination condition to the number of iterations required to achieved a given distance to the optimum (Zielinski and Laur, 2008). Although ex-

periments report promising results, the statistical tools used so far can not answer two main (still open) questions. How can we define a termination condition? Given a confidence level, how can we estimate an upper-bound for the number of iterations required to ensure convergence?

We propose posing the termination problem in statistical inference terms. From the perspective of statistical inference, the termination problem consists in designing a measure (depending only on the EA output) that correlates to the accuracy of the solution, so that they can be swapped. This paper introduces a general inference model for predicting the accuracy of the EA solution from the EA current state. We show that a linear regression model in logarithmic scale accurately relates accuracy and distribution-based quantities. We use the inference model to compare several types of distribution-based quantities reported in the literature (Zielinski and Laur, 2008). Our experiments indicate that the maximum distance to the best individual is the best choice in terms of computational efficiency and capability of predicting EA accuracy.

## Inference Model

All measures are taken in the domain of definition of the objective function, that is in the parameter space of the population being evolved. The distance to the (known) function minimum is our gold-standard reference convergence criterion, given that is directly associated to the algorithm accuracy. This criterion can only be computed if the optimum of the test function is known and, thus, is useless in real-world problems. We compute it as the maximum distance to the function minimum of a certain percentage  $p$  of the individuals (Zielinski and Laur, 2008) and denote it by *RefCrit*. Regarding the alternative quantities, which we will denote by *AltCrit* in general, we have considered the following distribution-based quantities (Zielinski and Laur, 2008):

1. **Maximum Distance (MxD).** It is given by the maximum distance of the population to the best individual.
2. **Population Variability (Std).** It is the maximum of the standard deviations (computed using the population individuals) of each dimension of the search space (in our case, the number of dimensions is limited to two).

Both quantities can be computed using all individuals or considering only a percentage  $p$  of the individuals. The latter is computationally faster and will be indicated by the suffix **Quick**.

Our final goal is to control (predict) the values taken by *RefCrit* from the values taken by the alternative measure *AltCrit*. In inference statistics, this is achieved by relating both quantities using a regression model. From now on and when appropriate,  $x$  stands for *AltCrit* (explicative variable) and  $y$  for *RefCrit* (response variable).

## Regression Model

Given a sampling of two random variables ( $x$  and  $y$ ), the linear regression of  $y$  (response variable) over  $x$  (explicative variable) is formulated as:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad (1)$$

where  $x_i, y_i$  are the sampling of  $x$  and  $y$  and  $\varepsilon_i$  a random error satisfying:

### Model Assumptions

1. Linearity:  $E(\varepsilon_i) = 0$
2. Homocedasticity:  $VAR(\varepsilon_i) = \sigma^2, \forall i$
3. Uncorrelation:  $COV(\varepsilon_i \varepsilon_j) = 0, \forall i, j$
4. Gaussianity:  $\varepsilon_i \sim N(0, \sigma^2)$ , for  $N(0, \sigma^2)$  a normal distribution.

The parameters of the regression model (1) are the regression coefficients  $\beta = (\beta_0, \beta_1)$  and the error variance  $\sigma^2$ . The regression coefficients describe the way the two variables relate, while the variance indicates the accuracy of the model and, thus, measures to what extent  $x$  can predict  $y$ .

Given that, in our case, the inference is over *RefCrit*, our model is:

$$RefCrit_i = \beta_0 + \beta_1 AltCrit_i + \varepsilon_i \quad (2)$$

for *RefCrit<sub>i</sub>*, *AltCrit<sub>i</sub>* the values of *RefCrit* and *AltCrit* obtained at the  $i$ -th iteration.

For a sample of length  $N$  (in our case  $N$  is the number of iterations), the regression coefficients,  $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1)$ , are computed by Least Squares Estimation (LSE) as:

$$\hat{\beta} = (X^T X)^{-1} X^T Y \quad (3)$$

for  $X = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_N \end{pmatrix}$ ,  $Y = (y_1, \dots, y_N)$  and  $^T$  denoting the transpose of a matrix. The differences between the estimated responses,  $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ , and the observed responses  $y_i$ :

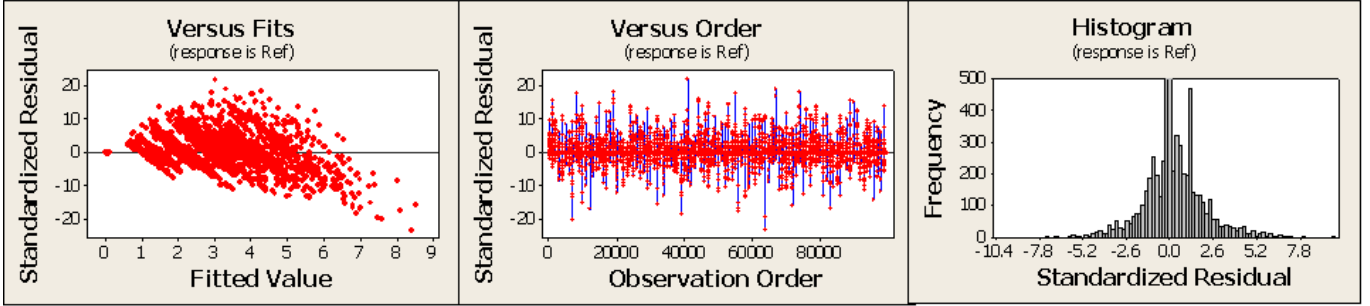
$$e_i = y_i - \hat{y}_i$$

are called residuals. Their square sum provides an estimation of the error variance:

$$S_R = \hat{\sigma}^2 = \frac{\sum e_i^2}{N - 2} \quad (4)$$

The four model conditions endow desirable properties to the LSE of the regression coefficients (Ashish, 1990). By the Gauss-Markov theorem under the first three assumptions, the LSE are best linear unbiased estimators and assure that predictions made by least squares fitted equations are

## Residual model diagnostics for normal scale



## Residual model diagnostics for logarithmic scale

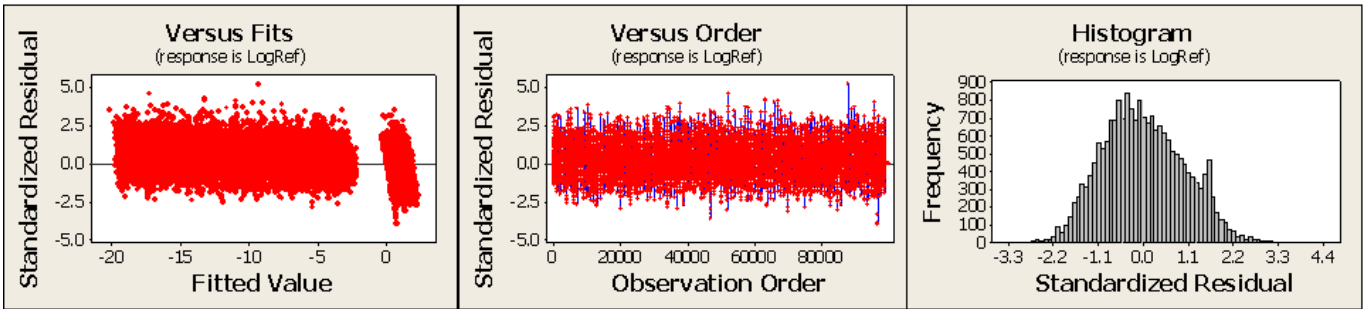


Figure 1: Residual diagnosis plots for Rastrigin function

good. By adding the fourth assumption (error gaussianity), the LSE has minimum variance among all unbiased estimators (not just linear) and allows the use of parametric tests, such as the Student's t-test for testing hypothesis on parameter values. The central limit theorem (asymptotically) guarantees this last property for large samples. Therefore, given that we have as many samples as EA iterations, in our case, the gaussianity is not a critical issue.

The standardized residuals:

$$en_i = (e_i - \mu(e_i)) / \text{std}(e_i)$$

, for  $\mu$  the average and  $\text{std}$  the standard deviation, are used to verify the model assumptions. The plot of  $en_i$  over  $\hat{y}_i$  is called the versus fit plot and reflects linearity (in the measure that it is centered at zero) and homocedasticity (uniform deviation from zero). The plot of  $en$  vs the sorted explicative variable is called the versus order plot and serves to detect any correlation pattern. Finally, the histogram of the standardized residuals reflects Gaussianity (Newbold et al., 2007).

Figure 1 shows the residuals diagnosis plots for Rastrigin function. From left to right, we plot the versus fit plot, the versus order plot and the histogram for the standardized residuals  $en$ . The plots at normal scale in the first row show that linearity (versus fit plot is centered at zero) and uncorrelation (versus order plot presents no pattern) are fully satisfied. Meanwhile, we observe a clear heteroscedasticity in the versus fit plot which presents an increasing deviation from zero.

This heteroscedasticity is due to a decrease in the population sparseness at advance stages of EA and also affects the Gaussianity assumption, as shown in the histogram of the first row. A monotonous increase in  $\sigma^2$  is usually solved by taking logarithms in both variables (Arnold, 1997). The residuals plots for the regression model in logarithmic scale (second row in fig. 1) indicate a good homocedasticity and Gaussianity for the standardized residuals.

From now on, the values of  $RefCrit$  and  $AltCrit$  will be assumed to be in logarithmic scale for the inference model:

$$\log(RefCrit_i) = \beta_0 + \beta_1 \log(AltCrit_i) + \varepsilon_i \quad (5)$$

We note that, by taking exponentials, the regression model in the original scale is polynomial with multiplicative errors:

$$RefCrit_i = e^{\beta_0} AltCrit_i^{\beta_1} e^{\varepsilon_i} \quad (6)$$

**Model verification** Previous to any kind of inference, it is mandatory to verify that the estimated parameters make sense. That is, whether it really exists a linear relation between  $x$  and  $y$ . By the Gauss-Markov theorem, such linear relation can be statistically checked using the following T-test (Newbold et al., 2007)

$$H_0 : \beta_1 = 0 \quad H_1 : \beta_1 \neq 0$$

where a  $p$ -value close to zero (below  $\alpha$ ) ensures the validity of the linear model with a confidence  $(1 - \alpha)100\%$ .



## Prediction Model

In order to predict the values of *RefCrit* from the values achieved by *AltCrit*, we use the regression prediction intervals (Newbold et al., 2007):

$$PI(x_0) = [L_{PI}(x_0), U_{PI}(x_0)]$$

since, for each  $x = x_0$ , they provide ranges for  $y$  at a given confidence level  $1 - \alpha$ . That is, given  $x_0$ , the values of the response  $y$  are within  $L_{PI}(x_0) \leq y \leq U_{PI}(x_0)$  in  $(1 - \alpha)100\%$  of the cases.

Given  $x_0 = AltCrit_0$ , the confidence interval at a confidence level  $(1-\alpha)$  predicting *RefCrit* is given by:

$$\begin{aligned} PI(x_0) &= [L_{PI}(x_0), U_{PI}(x_0)] = \\ &= [\hat{y}_0 - t_{\alpha/2} S_R \sqrt{1 + h_0}, \hat{y}_0 + t_{\alpha/2} S_R \sqrt{1 + h_0}] \quad (7) \end{aligned}$$

for  $t_{\alpha/2}$  the value of a T-Student distribution with  $N - 2$  degrees of freedom having a cumulative probability equal to  $\alpha/2$  and:

$$\begin{aligned} \hat{y}_0 &= b_0 + b_1 x \\ h_0 &= (1 \quad x_0)(X^T X)^{-1} \begin{pmatrix} 1 \\ x_0 \end{pmatrix} = a_0 + a_1 x_0 + a_2 x_0^2 \end{aligned}$$

Where  $(a_0, a_1, a_2)$  stand for the coefficients of the quadratic polynomial resulting from the previous algebraic expression.

The exponential of *PI* already provides (with confidence  $1 - \alpha$ ) an upper bound for the accuracy of EA solution given EA current state. In order to obtain the upper bound for *AltCrit* ensuring a given accuracy  $U_{PI}(x_0)$ , it suffices to find the value  $x_0$  that solves:

$$\hat{y}_0 + t_{\alpha/2} S_R \sqrt{1 + h_0} = U_{PI}(x_0) \quad (8)$$

Using the expressions for  $\hat{y}_0$  and  $h_0$  in (8) and solving for  $x_0$ , we obtain:

$$x_0 = \frac{2b_0b_1 - t_{\alpha/2}^2 S_R^2 a_1 - 2b_1 U_{PI}(x_0) + \sqrt{D}}{2(t_{\alpha/2}^2 S_R^2 a_2 - b_1^2)} \quad (9)$$

where the discriminant is given by:

$$\begin{aligned} D &= (t_{\alpha/2}^2 S_R^2 a_1 - 2b_0b_1 + 2b_1 U_{PI}(x_0))^2 - \\ &- 4(t_{\alpha/2}^2 S_R^2 a_2 - b_1^2)(t_{\alpha/2}^2 S_R^2 (a_0 + 1) - \\ &- b_0^2 + 2U_{PI}(x_0)b_0 - U_{PI}(x_0)^2) \end{aligned}$$

By taking exponentials from (9) we get the upper bound for *AltCrit*.

## Experimental Settings

In this study we have compared the predictive capability of the following distribution-based measures given at the beginning of the previous Section: **MxD**, **MxDQuick** and

**StdQuick**. We have used  $p = 30\%$  of the population for computing **Quick** scores. We have considered six well-known test functions (Digalakis and Margaritis, 2002) having a minimum value of zero:

1. Esphere:

$$f_1(x) = \sum_{i=1}^n x_i^2$$

2. Rosenbrock:

$$f_2(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

3. Rastrigin:

$$f_3(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i + 10)]$$

4. Ackley:

$$f_5(x) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - e^{\frac{1}{2}\sum_{i=1}^n \cos(2\pi x_i)}$$

5. GoldstenPrice:

$$\begin{aligned} f_6(x) &= (1 + (x_1 + x_2 + 1)^2) \cdot \\ &\cdot (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \cdot \\ &\cdot (30 + (2x_1 - 3x_2)^2) \cdot \\ &\cdot (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2) \end{aligned}$$

6. Easom:

$$\begin{aligned} f_7(x) &= -\cos x_1 \cdot \cos x_2 \cdot \\ &\cdot \exp(-((x_1 - \pi)^2 + (x_2 - \pi)^2)) \end{aligned}$$

We have used a Differential Evolution (DE) technique for the minimization task. Differential evolution is a real parameter encoding evolutionary algorithm for global optimization over continuous spaces (Storn and Price, 1997; Das and Konar, 2005). In this paper, we use the 3-parameter DE1 scheme (Storn and Price, 1997) for solving DE. For a real search space of dimension  $D$ , the population is randomly initialized with  $ND$  vectors (for  $ND$  the first algorithm parameter). Each vector  $v$  in the population is evolved by mutation and recombination operators. Given a mutation rate  $F \in [0, 2]$  (second parameter of the algorithm), the mutation operator produces a new vector  $vm$  by adding a vector difference of two randomly chosen population vectors  $v1$  and  $v2$  to another randomly chosen vector  $v3$ :

$$vm = v1 + F(v2 - v3)$$

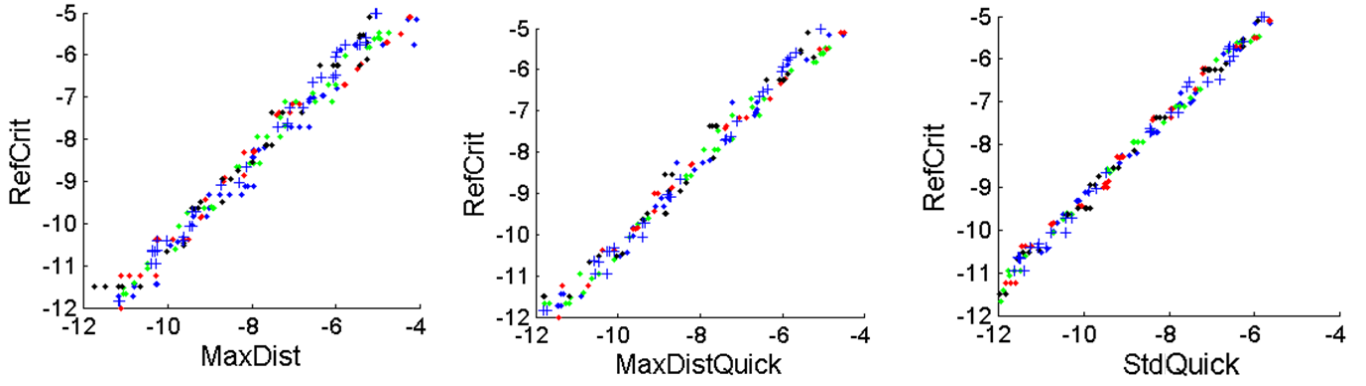


Figure 2: Scattered plots for Rastrigin test function and 10 different runs of DE in logarithmic scale.

For the recombination step, a new vector  $vf$  is created from the mutation vector by means of a combination rate  $CR$  (third parameter of the algorithm) as follows:

$$vf_i = \begin{cases} vm_i & \text{if } r_i < CR \text{ or } i = k \\ v_i & \text{otherwise} \end{cases}$$

for  $vf_i$  the  $i$ -th component of  $vf$  and  $r_i \in [0, 1]$  a random number and  $k$  a random number uniformly distributed in  $[1, D]$ . Finally a selection operator is applied. The vector  $vf$  and the initial vector  $v$  are compared and the vector that better fits the objective function is selected and remains in the next population. This process is iteratively repeated until a stopping criterion is reached. Following the literature (Das and Konar, 2005), we have chosen the following values for DE parameters:  $D=2$ ,  $ND=20$ ,  $F=0.9$ ,  $CR=0.5$ . For each test function, we have executed 100 runs of the algorithm during 10.000 iterations each one.

For each test function and alternative quantity, two different experiments have been carried out:

1. **Model Assessment.** The suitability and accuracy of the linear model in logarithmic scale has been assessed by the T-test on the regression coefficients, as well as, the analysis of the residuals variance ( $S_R$ ).
2. **Model Prediction.** In order to assess the prediction capabilities of each model two different experiments have been addressed. On one hand, we have explored the relation between  $RefCrit$  and  $AltCrit$  by analyzing the confidence intervals of the regression coefficients. On the other hand, we have compared the prediction intervals across the three distribution-based quantities.

## Experiments and Results

### Model Assessment

Figure 2 shows scattered plots associated to the regression model for the Rastrigin test function. The  $y$  axis represents

$RefCrit$  values and the  $x$  axis each of the alternative quantities (from left to right **MxD**, **MxDQuick** and **StdQuick**). Each plot shows 10 different runs marked with distinct colors and markers. For all alternative quantities, we observe a uniform behavior across DE executions, which present the same linear pattern with a small variation.

Table 1 reports the estimation of the model parameters (the regression coefficients  $\hat{\beta}_0$ ,  $\hat{\beta}_1$  and the residual variance  $S_R$ ) and the p-value of the model verification T-test. We report values for each test function (rows) and alternative quantity (columns). For all cases, there is a clear linear relation between accuracy and the alternative quantities (with p close to the working precision). Besides the goodness-of-fit is excellent, given that  $S_R$  is extremely small compared to the variable ranges (see fig. 2).

Concerning the relation between the two variables, it is worth noticing two aspects. Firstly, we observe that the estimated slope  $\hat{\beta}_1$  is close to 1 for all cases. This implies that the relation in logarithmic scale is a translation of the identity and the regression model in the original scale is also linear. Secondly, the constant coefficients  $\hat{\beta}_0$  are sorted as follows:

$$\hat{\beta}_0(\text{MxD}) \leq \hat{\beta}_0(\text{MxDQuick}) \leq 0 \leq \hat{\beta}_0(\text{StdQuick})$$

The above commented points indicate that there might be the following tendency:

$$\text{StdQuick} \leq RefCrit \leq \text{MxDQuick} \leq \text{MxD}$$

This already suggests that the value of maximum distances itself might guarantee an upper bound for the EA accuracy. In order to really confirm such hypothesis, we should analyze the prediction intervals.

### Model Prediction

Figure 3 shows the prediction intervals for the 6 test functions. Each plot shows the prediction interval for all alternative quantities, as well as the identity line (solid line)

	MxD				MxDQuick				StdQuick			
	$p$	$b_1$	$b_0$	$S_R$	$p$	$b_1$	$b_0$	$S_R$	$p$	$b_1$	$b_0$	$S_R$
<b>Esfere</b>	$\leq 10^{-32}$	1.000	-0.487	0.12	$\leq 10^{-32}$	1.000	-0.183	0.05	$\leq 10^{-32}$	1.000	0.685	0.03
<b>Rosenbrock</b>	$\leq 10^{-32}$	1.002	-0.478	0.16	$\leq 10^{-32}$	1.002	-0.122	0.09	$\leq 10^{-32}$	1.003	0.761	0.05
<b>Rastrigin</b>	$\leq 10^{-32}$	0.996	-0.557	0.07	$\leq 10^{-32}$	1.006	-0.207	0.06	$\leq 10^{-32}$	1.007	0.686	0.03
<b>Ackley</b>	$\leq 10^{-32}$	1.000	-0.487	0.07	$\leq 10^{-32}$	1.004	-0.148	0.06	$\leq 10^{-32}$	1.004	0.725	0.03
<b>GoldstenPrice</b>	$\leq 10^{-32}$	0.998	-0.504	0.16	$\leq 10^{-32}$	1.001	-0.127	0.07	$\leq 10^{-32}$	1.002	0.804	0.05
<b>Easom</b>	$\leq 10^{-32}$	1.001	-0.484	0.05	$\leq 10^{-32}$	1.011	-0.143	0.04	$\leq 10^{-32}$	1.011	0.766	0.03

Table 1: Model fitting scores

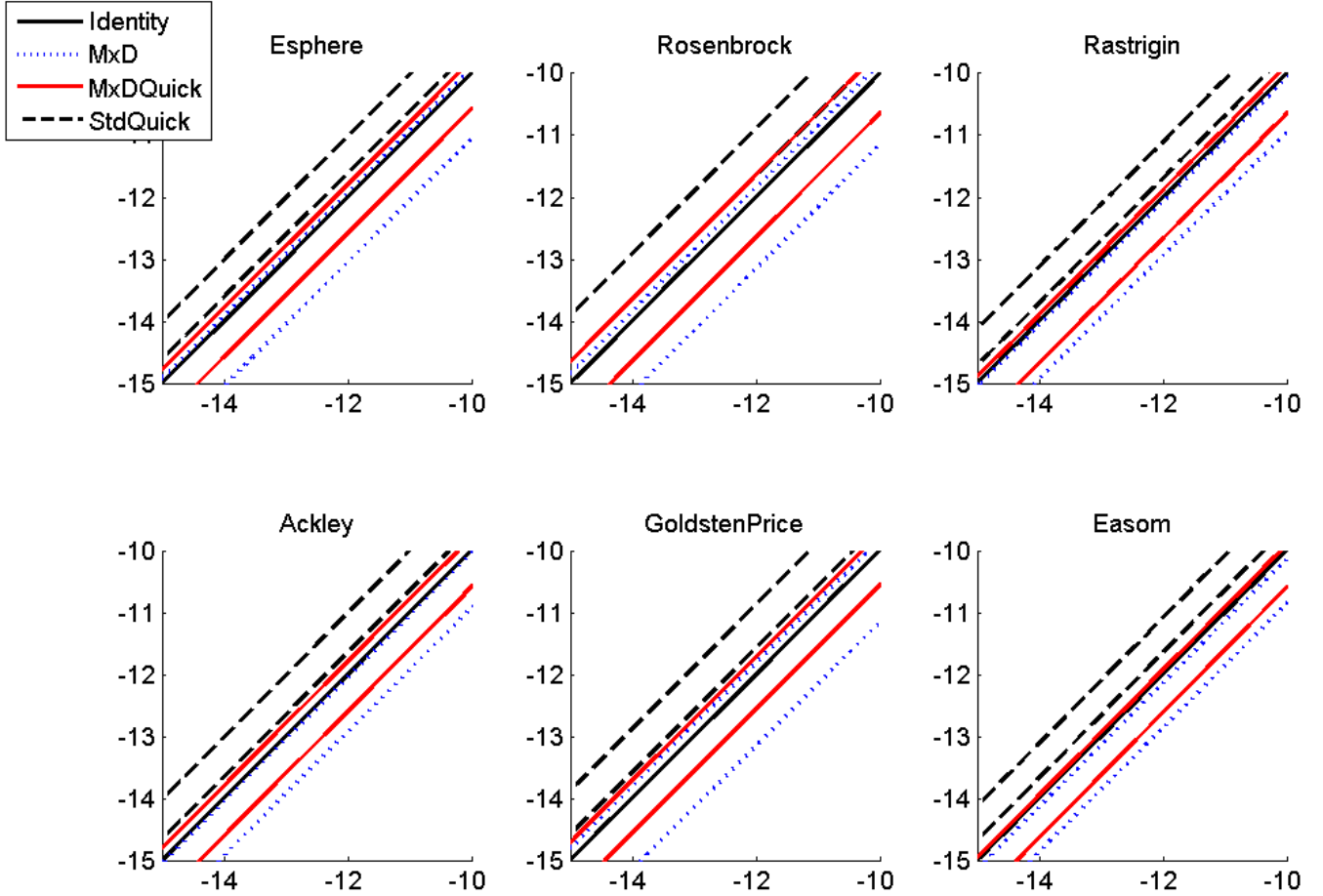


Figure 3: Prediction intervals

for a better visual comparison between *AltCrit* prediction and *RefCrit* values. The alternative quantity can substitute *RefCrit* in the measure that the identity line is within the range given by the prediction interval. This is the case for quantities based on maximum distances. In the case of **StdQuick** the predicted values are above the reference identity line. This implies that **StdQuick** and *RefCrit* can not be directly swapped and, thus, we need the upper bound given in (9) for predicting *RefCrit* values.

Table 2 reports the upper bounds for each alternative quantity ensuring a given accuracy for *RefCrit*. For each test function (rows), we report values for two accuracies  $10^{-6}$  and  $10^{-9}$ . As suggested by the plots in fig. 3, for Rastrigin, Ackley and Easom test functions, the upper bound for **MxD** is almost equal to the accuracies  $10^{-6}$  and  $10^{-9}$ . This is also the case for Easom test function and **MxDQuick**. For the remaining cases, **MxD** and **MxDQuick** upper bounds are a little lower (though still comparable). We would like

		<b>MxD</b>	<b>MxDQuick</b>	<b>StdQuick</b>
<b>Esfere</b>	$10^{-6}$	$8.3 \cdot 10^{-7}$	$7.5 \cdot 10^{-7}$	$3.5 \cdot 10^{-7}$
	$10^{-9}$	$8.3 \cdot 10^{-10}$	$7.5 \cdot 10^{-10}$	$3.5 \cdot 10^{-10}$
<b>Rosenbrock</b>	$10^{-6}$	$7.5 \cdot 10^{-7}$	$6.4 \cdot 10^{-7}$	$3.1 \cdot 10^{-7}$
	$10^{-9}$	$7.6 \cdot 10^{-10}$	$6.5 \cdot 10^{-10}$	$3.2 \cdot 10^{-10}$
<b>Rastrigin</b>	$10^{-6}$	$9.8 \cdot 10^{-7}$	$8.4 \cdot 10^{-7}$	$4.0 \cdot 10^{-7}$
	$10^{-9}$	$9.5 \cdot 10^{-10}$	$8.7 \cdot 10^{-10}$	$4.2 \cdot 10^{-10}$
<b>Ackley</b>	$10^{-6}$	$9.7 \cdot 10^{-7}$	$7.6 \cdot 10^{-7}$	$3.5 \cdot 10^{-7}$
	$10^{-9}$	$9.7 \cdot 10^{-10}$	$7.8 \cdot 10^{-10}$	$3.6 \cdot 10^{-10}$
<b>GoldstenPrice</b>	$10^{-6}$	$7.3 \cdot 10^{-7}$	$7.0 \cdot 10^{-7}$	$3.0 \cdot 10^{-7}$
	$10^{-9}$	$7.2 \cdot 10^{-10}$	$7.0 \cdot 10^{-10}$	$3.1 \cdot 10^{-10}$
<b>Easom</b>	$10^{-6}$	$1.0 \cdot 10^{-6}$	$8.9 \cdot 10^{-7}$	$3.9 \cdot 10^{-7}$
	$10^{-9}$	$1.1 \cdot 10^{-9}$	$9.6 \cdot 10^{-10}$	$4.2 \cdot 10^{-10}$

Table 2: Upper-bound ensuring a given accuracy of EA

to note that this does not contradict the swapability of the two quantities. The upper bound condition requires that  $RefCrit$  equals  $U_{PI}$ . This is a stronger condition than the swapability one, which just requires  $RefCrit \leq U_{PI}$ . Concerning **StdQuick**, its upper bounds are clearly lower (a 40% at most) than the two accuracies. This confirms that **StdQuick** and  $RefCrit$  are not directly swapable.

## Conclusions and Future work

In real-world problems (which have unknown optimums) it is mandatory to design a termination condition for EA ensuring stabilization close to the unknown optimum. As far as we know, this is the first work addressing EA termination condition in terms of statistical inference. In this context, we have explored to what extent a reference quantity (not available in real-world problems) measuring EA accuracy ( $RefCrit$ ) can be substituted by an alternative quantity ( $AltCrit$ ) computed from EA population.

According to our experiments on several known test functions, there is a strong (almost ideal) linear relation between distribution-based quantities (**MxD**, **MxDQuick** and **StdQuick**) and the distance to the optimum. This allows analyzing the prediction capabilities of each distribution-based quantity by means of the regression prediction intervals. From our analysis, we conclude that quantities based on maximum distances (**MxD**, **MxDQuick**) have the highest concordance to EA accuracy and, thus, can substitute it as termination condition. Given that **MxDQuick** is computationally faster than **MxD**, it is the best candidate for terminating EA in real-world problems.

We consider that there are some issues that should be further developed. The test functions used are a small set of benchmarking data sets (we cover two out of the five categories described in (Hansen et al., 2010)) and only 2-D problems have been solved. However, the functions used include three properties (multimodality, global structure and scalability) reported in a recent study (Mersmann et al., 2010)

to have a high influence in the performance of EA's. In order to fully test the applicability to real-world problems, we will enlarge the test set to include groups of functions with specific key features (Hansen et al., 2010) affected by noise and stochastic variability. Regarding size, although it definitely influences convergence rate (more iterations of EA are required (Hansen et al., 2010)), this is independent of the relationship between  $RefCrit$  and  $AltCrit$ . Thus, size is not a limitation for the prediction model, which links convergence rate with population stability.

In this study we have restricted to DE algorithm. We are currently extending our analysis to other EA methods in order to cover existing EA paradigms: genetic algorithms (Goldberg and Richardson, 1987), evolutionary strategies (Beyer and Schwefel, 2002), particle Swarm optimization (Barrera and Coello, 2009), among others. Nevertheless, we do not expect any significant changes in our conclusions since DE already presents the main features of EA (Ronkkoenen, 2009).

In our experimental setting test functions have been studied separately. We consider that the influence of the test function should be taken into account, so that the inference can be done independently of the function features. This will be studied by using generalized regression models including random effects (Lee et al., 2006) modelling the impact of the test function group.

Finally, it is worth noting that in numerical analysis, a termination condition for an iterative scheme only makes sense in the case that the algorithm converges (that is, it reaches a steady point). The convergence rate of an iterative minimizing method depends on some properties of the target function (whether there is a minimum or not) and the method itself (its capability to find the minimum). In complex real-world problems, there is no guarantee that such conditions will be satisfied. Therefore, in practice, a termination condition in terms of a number of iterations or function evaluations is required in order to guarantee that the algorithm

stops. We note that this is not a specific limitation of our methodology, but a general feature of real-world applications, which might present a poor convergence rate.

**Acknowledgements.** This work was supported by the Spanish projects PI071188, TIN2009-13618 and CONSOLIDER-INGENIO 2010 (CSD2007-00018). The 2nd author has been supported by The Ramon y Cajal Program and SAF2010-19257, Fundació La Marató de TV3 (070530).

We would like to thank Llorenç Badiella from the Statistics Service at the Universitat Autònoma de Barcelona for his advice.

## References

- Arnold, S. (1997). *The theory of linear models and multivariate observations*. Wiley.
- Ashish, S. and Srivastava, M. (1990). *Regression analysis. Theory, methods and applications*. Springer-Verlag.
- Bäck, T., Hammel, U., and Schwefel, H. (1997). Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, 1(3):1–17.
- Barrera, J. and Coello, C. A. C. (2009). *A Review of Particle Swarm Optimization Methods used for Multimodal Optimization*, pages 9–37. Springer-Verlag.
- Beyer, H.-G. and Schwefel, H.-P. (2002). Evolution strategies: A comprehensive introduction. *Journal Natural Computing*, 1(1):3–52.
- Cagnoni, S., Lutton, E., and Olague, G. (2008). *Genetic and Evolutionary Computation for Image Processing and Analysis*. Hindawi Publishing Corporation.
- Das, S. and Konar, A. (2005). An improved differential evolution scheme for noisy optimization problems. In *Pattern Recognition and Machine Intelligence, LNCS Volume 3776*, pages 417–421. Springer.
- Digalakis, J. and Margaritis, K. (2002). An experimental study of benchmarking functions for genetic algorithms. *International Journal of Computer Mathematics*, 79(4):403–416.
- Goldberg, D. and Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In *Second International Conference on Genetic Algorithms and their Application*, pages 41–49.
- Hansen, N., Auger, A., Ros, R., Finck, S., and Posík, P. (2010). Comparing results of 31 algorithms from the black-box optimization benchmarking. In *Proc. of GECCO 2010*.
- Holland, J. (1975 (1992 edition)). *Adaptation in Natural and Artificial Systems*. MIT Press.
- Lee, Y., Nelder, J., and Pawitan, Y. (2006). *Generalized linear models with random effects*. Chapman and Hall/CRC Boca Raton.
- Mersmann, O., Preuss, M., and Trautmann, H. (2010). Benchmarking evolutionary algorithms: Towards exploratory landscape analysis. In *PPSN XI, Part I, LNCS 6238*, pages 73–82.
- Newbold, P., Carlson, W., and Thorne, B. (2007). *Statistics for Business and Economics*. Pearson Education, 6th edition.
- Price, K., Storn, R., and Lampinen, J. (2005). *Differential Evolution: A Practical Approach to Global Optimization*. Springer.
- Ravikumar, V. and Panigrahi, B. (2010). Comparative study of evolutionary computing methods for parameter estimation of quality signals. *International Journal of Applied Evolutionary Computation*, 1 (2):28–59.
- Rizki, M., Zmuda, M., and Tamburino, L. (2002). Evolving pattern recognition systems. *IEEE Transactions on Evolutionary Computation*, 6 (6):594–609.
- Ronkkonen, J. (2009). *Continuous Multimodal Global Optimization with Differential Evolution Based Methods*. PhD thesis, Lappeenranta University of Technology.
- Safe, M., Carballido, J., and Ponzoni, I. (2004). On stopping criteria for genetic algorithms. In *Advances in Artificial Intelligence, SBIA 2004, LNCS, Volume 3171*, pages 405–413.
- Schwefel, H. (1995). *Evolution and Optimum Seeking*. Wiley, New York.
- Storn, R. and Price, K. (1997). Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359.
- Tagetiren, M. and Suganthan, P. (2006). A multi-populated differential evolution algorithm for solving constrained optimization problem. In *IEEE Congress on Evolutionary Computation*, pages 340–347.
- Zielinski, K. and Laur, R. (2008). Stopping criteria for differential evolution in constrained single-objective optimization. In *Advances in Differential Evolution*, volume 143 of *Studies in Computational Intelligence*, pages 111–138. Springer Berlin / Heidelberg.

# Local Information Maximisation creates Emergent Flocking Behaviour

Christoph Salge<sup>1</sup> and Daniel Polani<sup>1</sup>

<sup>1</sup>University of Hertfordshire, Hatfield, UK  
c.salge@herts.ac.uk

## Abstract

The three boids rules of alignment, separation and cohesion, introduced by Reynolds to recreate flocking behaviour have become a well known standard to create swarm behaviour. In this paper we want to demonstrate how similar flocking behaviour can be created by a local, agent based model, following a principle of information maximisation. The basis for our model is an extension of Vergassola's *infotaxis* model, where agents determine their actions based on the highest expected reduction of entropy. We adapted this approach to a grid world-based search task, and extended the agents abilities so they could not only perform a Bayesian update with information gained from the environment, but also with information gained from other agents. The resulting global flocking behaviour is then analysed in regard to how well it resembles the basic boids rules.

## Introduction

Flocking behaviour is a natural phenomenon found in a diverse selection of life forms, such as birds, fish, herd animals and insects. And, as demonstrated by Dyer et al. [8], in specific circumstances even humans exhibit similar behaviour. One of the first models to create this behaviour in a computer simulation is the *boids* steering model, introduced by Reynolds [14]. The model is a prime example of a powerful artificial life idea, namely how local self organisation can create emergent global phenomena. Originally developed to animate the movement of fish and birds for graphical presentation, the boids model has developed into a "de facto" standard for flocking algorithms.

The three basic rules, alignment, separation and cohesion, are agent based and local, so they allow every agent to determine its own actions by itself, using only local data:

- Alignment: steer towards the average heading of local flock mates
- Separation: steer to avoid crowding local flock mates
- Cohesion: steer towards the average position of local flock mates

This model, or variations thereof, are not only the basis for many current flocking and swarm simulations, but are also a powerful example for how simple, local rules can lead to the emergence of complex, life-like properties.

What we want to probe further in this paper is how the global phenomenon of self-organised flocking can be explained; but instead of motivating the individual atomic rules, we intend to challenge the notion that those rules are necessarily atomic. As an alternative, we offer a model where the individual agent's actions, and the resulting global flocking behaviour, is created and motivated by obtaining as much relevant information about the environment as possible. This is an additional result of our previous efforts to extend information theoretic-behaviour generation in general, and in particular the biologically inspired *infotaxis* model by Vergassola et al. [25], to a multiagent system. In the original model the sensor inputs from the environment are used, via a Bayesian Update, to update an internal probabilistic model about a specific location. Actions are chosen based on how much expected information gain they provide for the internal model. In the multiagent model, the actions of other, observable agents are treated with the same Bayesian update, and the resulting agent movement starts to resemble flocking behaviour.

In this paper we are first going to describe our model, and how the single principle of maximal information gain can be used to generate agent behaviour. We shall then demonstrate how information, both from the environment and from other agents, is integrated into the Bayesian model of the agent. The resulting behaviour of those models is then analysed by measuring how well it resembles certain basic characteristics of boids flocking behaviour. We also offer a less formal explanation on how the mechanism of information maximisation leads to flocking behaviour, and how this could be generalised.

## Related Work

Information Theory was originally conceived by Shannon [17] to deal with the limits of transatlantic communication; the main focus being the optimal use of a limited commu-

nication channel. But its considerable mathematical versatility, since it can be applied on any system that can be formalised in terms of random variables, also allows for the analysis of a diverse variety of systems in terms of their information theoretic properties and limitations [5].

A recent information theoretic analysis of a boids-like swarm model by [4] demonstrated the ability for information transfer between the flocking agents. Few “informed” agents were capable of steering a swarm. Corresponding results have been observed in the flocking behaviour of human crowds by [8]. If we take a closer look at coordinated systems in nature, it is not surprising that there is a certain degree of mutual information between the organised components. Organisation requires a certain degree of causal dependence, and if we follow the argument of [12], this leads to a certain degree of mutual information between the appropriate variables. Similar conclusion can be drawn for the necessity of information flow, as defined in [2]. The mere presence of some non-vanishing correlation, i.e. nonzero mutual information in nature is, of course, not surprising. However, it is striking that there are many indications that biological organisms tend indeed to operate close to the physical limits for sensory and informational capacities [11, 15]. This can be formulated as an information optimality principle which provides a constructive way to generate behaviours. The use of information theory to model the complexity of cognitive processes [18, 21] has lead to systematic approaches to model agent decision making [22, 23, 6] utilizing information theory in a constructive way, beyond the use as a merely analytic tool. To mention a few examples; it has been used to optimize behaviour in a Reinforcement Learning-like context by [20]. Also, for behaviour generation, there is the predictive information maximization [1] which is related to the dynamical systems *homeokinesis* principle by [7].

Another example is the idea of *empowerment* by [9], where an agent tries to act as to maximise the channel capacity between its actuators and sensors which essentially is an optimization of its sensorimotor niche. [3] demonstrates that this principle on its own can already leads to coordinated multiagent behaviour. Note that this shows how, seemingly in opposition to the original philosophy behind information theory which had been designed to carry no semantics, our current work is based on ideas that one is able to distinguish between relevant and non-relevant information.

The *information bottleneck* perspective by [19] demonstrates how the notion of Shannon information can be imbued with relevance, and this can be achieved either through the presence of goals or reward structures [13, 24] or, alternatively, imprinted by the agent-environment interaction itself [10].

This concept of *relevant information* [13] is one we refer to when we later talk about an agent maximising information. *Relevant information* is interpreted here according to the information bottleneck formalism [19, 13]. It quantifies

not *all* information (i.e. possible reduction of uncertainty) in the environment, but only that information which identifies the selection of optimal actions by the agent. Under this perspective, any information in the environment beyond that is ignored.

## Information Theory

We consider random variables  $X$  which can assume concrete values  $x$ . Write  $P(X = x)$ , or  $p(x)$  by abuse of notation, for the probability of  $X$  assuming the specific value  $x$ . We can now define the entropy  $H(X)$  of the random variable  $X$  as

$$H(X) = - \sum_x P(X = x) \cdot \log P(X = x) \quad (1)$$

This is often used to describe the uncertainty about the outcome of  $X$ . An alternative, equivalent interpretation is to consider  $H(X)$  as the average expected “surprise” or the information gained if one was to observe the state of  $X$ , if all one knows about  $X$  is only its distribution  $P(X)$ .

The entropy has a number of important properties. Among others, as it is an *a priori* uncertainty, the entropy is larger if the outcomes are more evenly distributed than if the outcomes are more concentrated on a particular value — in other words, concentrated values are easier to predict (and less uncertain) than uniformly spread ones.

Consider now two jointly distributed random variables,  $X$  and  $Y$ ; then we can calculate the conditional entropy of  $X$  given a particular outcome  $Y = y$  as:

$$H(X|Y = y) = \sum_x P(X = x|Y = y) \cdot \log P(X = x|Y = y) \quad (2)$$

This can also be generalised to the entropy of  $X$ , given the random variable  $Y$  in general, and is obtained by averaging over all possible outcomes of  $Y$ :

$$H(X|Y) = \sum_y p(y) \cdot H(X|Y = y) \quad (3)$$

This is the entropy of  $X$  that remains if  $Y$  is known. Finally, consider  $H(X)$  and  $H(X|Y)$ , the entropy of  $X$  before and after we learn the state of  $Y$ . Thus, their difference is the amount of information we can learn about  $X$  by knowing  $Y$ . Subtracting one from the other, we get a value called mutual information:

$$I(X; Y) = H(X) - H(X|Y) \quad (4)$$

This is the value we will refer to if we use the term information and it is measured in bits; if one variable is said to have information about another it means that the mutual information between them is non-zero. As the mutual information is symmetrical ([5]), this works both ways, so one variable  $A$  contains as much information about  $B$ , as  $B$  does about  $A$ .

Importantly, note that this original notion of information does not include any semantics and only depends on the joint



distribution of  $X$  and  $Y$ . Therefore, in calculating the entropy and the mutual information measures, the labels of the values of the variables are not relevant.

In the specific model described in the next section, we will maximise the information in respect to a specific location of a resource, but the model is entirely general and the information could correspond to any kind of information about the state of the agent's environment needed for the agent to increase its performance.

## Experimental Model

### Scenario

We consider a model consisting of agents situated in a torus-shaped grid world of size  $n \times m$  with periodic boundary conditions. Each location in this world is in the set  $\mathcal{W} = \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$ . There is one single location of interest  $F^*$ , defined also over the set  $\mathcal{W}$ . To contextualise, we will call the location the *food source*, but one can interpret it as any other relevant location information, such as position of shelter or mates. The goal of the agents is to determine (not reach) this location in the shortest possible time. The agents' initial location, and the location of the food are randomly generated at the start of the simulation, and each time step an agent can execute a move action which moves it one cell up, down, left or right. The agent then gets new sensor inputs; it is able to see the state of the world in all cells not more than  $r$  cells away from it. Its sensor signal for each cell is a two-state random variable that indicates either that those cells are empty or that they contain the (here unique) food source. After this observation, the agent decides where to move next. This behaviour is repeated until the agent finds the food.

Once the agent finds the food, the agent disappears. An agent that has disappeared does not block other agents, cannot be observed, and its behaviour is not taken into account for the statistical measurements. Note that the food source itself is unaffected from agents finding it.

The above scenario determines the basic properties of our setting. Now, as we are interested in flocking behaviour, for an effective evaluation, the simulation will be run continuously, so the agents have time to form a swarm. Thus, instead of reinitializing the simulation every time one or all agents find the food source, at each time step there is a 3 % chance that the food will be randomly relocated. In this case, all agents' internal model is reset, so they start a new search. Those agents which have disappeared because they found the food will also be put back into the world in the location they previously disappeared from. The purpose of this is to allow swarms that have already formed to continue their coordinated movement.

### Agent Behaviour

In our model, the agents determine their actions by using an internal probability distribution  $F$ , which stores informa-

tion about the world. This internal distribution implements a Bayesian model for the location of the food source. More precisely  $F$  is also defined over  $\mathcal{W}$ , and  $P(F = f)$  corresponds to the probability of the food source being in location  $f$ , given the agent's current information.

Initially, all cells have the same probability of  $\forall f \in \mathcal{W} : p(f) = 1/(n \cdot m)$ , since the agent has no information about the location  $f$ . However, as the agent moves around, it can observe different locations in  $\mathcal{W}$ , and discovers that some locations are either empty or contain the food source. If  $f$  contains the food, then  $p(f) = 1$ . If  $f$  is empty, then  $p(f) = 0$ .

In both cases the probabilities of the other locations are normalised accordingly, so the sum of probabilities is always one. This operation is functionally identical to actually performing a Bayesian update with the observable environmental random variables, namely, the food state of the cells within the agent's sensor range.

The remaining uncertainty of the agent about the location of the food source is reflected by the internal probability distribution and can be measured in terms of entropy  $H(F)$ , where  $F$  is the agent-internal random variable corresponding to the expected position of the food.

### Infotaxis Search

To generate the agent's behaviour, we adopt a greedy information gain-maximisation algorithm, called *Infotaxis* by [25]. Infotaxis was shown to provide a biologically plausible principle as to how a moth could use the very sparse information provided by their olfactory sensors to determine the source of pheromones inside a wide area. The main idea is to act in a way that increases the expected gain in information at each time step. We adapted the infotaxis approach for our discrete grid world scenario.

Infotaxis behaviour is generated by the following steps:

1. Determine which action  $a$  will likely lead to the largest reduction in entropy  $H(F)$ , the uncertainty regarding the position of the food source.
2. Take action  $a$  and update  $F$  with the resulting sensor input.
3. If  $H(F) > 0$ , then repeat from step 1.

In step 1 the agent has to determine the likely reduction of entropy based on  $F$ , the agent's current "knowledge" about  $F^*$ .

Depending on the position  $w \in \mathcal{W}$  of the agent, there is a set  $\mathcal{S} \subseteq \mathcal{W}$  of the locations that are visible to the sensor of the agent. The visible locations are those within the agent's sensor range, meaning they are  $r$  or less cells away from the agent's position. If the agent, starting from the current position, takes the action  $a$  from its set of available actions  $\mathcal{A}$ , it will enter a new state  $w_a$ . In this new state the agent can now sense a new set of locations, denoted by  $\mathcal{S}_a$ .

To calculate the expected entropy reduction of action  $a$ ,  $\Delta H(a)$ , two cases have to be considered. In the first case, the actual location of the food source  $f \in \mathcal{W}$  would be inside the newly observed set of positions  $\mathcal{S}_a$ , inside the sensor range after the action  $a$  was taken by the agent. The agent assumes that this occurs with the probability of

$$P(f \in \mathcal{S}_a) = \sum_{f \in \mathcal{S}_a} P(F = f) \quad (5)$$

in reference to the agent internal model  $F$ . In this case the agent's uncertainty after carrying out action  $a$ ,  $H(F_a)$  would be reduced to zero, and the reduction of entropy would be the difference  $H(F) - H(F_a) = H(F)$ .

In the other case, the location  $f$  of the food source is not in  $\mathcal{S}_a$ . This occurs with a probability of  $1 - P(f \in \mathcal{S}_a)$ . In that case, we have to calculate an updated probability distribution for  $F$ , called  $F_a$ . According to Bayes' rule,  $P(F_a = f) = 0$  for all  $f \in \mathcal{S}_a$ , the resulting probability for all observed, empty locations to contain the food source is zero, and the remaining locations are normalized accordingly by:

$$\forall(f \notin \mathcal{S}_a) : P(F_a = f) := \frac{P(F = f)}{\sum_{w \notin \mathcal{S}_a} P(F = w)} \quad (6)$$

This divides the remaining non-zero probabilities, by the sum of their probabilities, normalizing the overall sum of all probabilities to 1. This updated version of  $F_a$  can then be used to calculate the reduction of entropy in the second case, which is given by the difference  $H(F) - H(F_a)$ . If we put all this together, the expected reduction of entropy for taking action  $a$  is:

$$\Delta H(a) = P(F \in \mathcal{S}_a) \cdot H(F) + P(F \notin \mathcal{S}_a) \cdot (H(F) - H(F_a)) \quad (7)$$

To summarize, each step the agent selects the action  $a$  that maximises  $\Delta H(a)$ . If several actions lead to the same expected entropy reduction, the agent selects one of them at random. The sensors are then updated as described above, and this behaviour is repeated until the food source is located. Essentially, this behaviour implements a version of Vergassola et al.'s *infotaxis* search and we will refer to it as such in the subsequent text.

## Social Bayesian Update

Earlier studies of single agent *infotaxis* behaviour in [16] demonstrated that the agent's actions contain information about the food source location. If we look at Fig.2, we see how the probability of the food source location is distributed conditioned on an agent moving north. More importantly, every agent which has to take in (a minimum amount of) relevant information to attain a certain performance level also *must necessarily* encode at least that amount of information

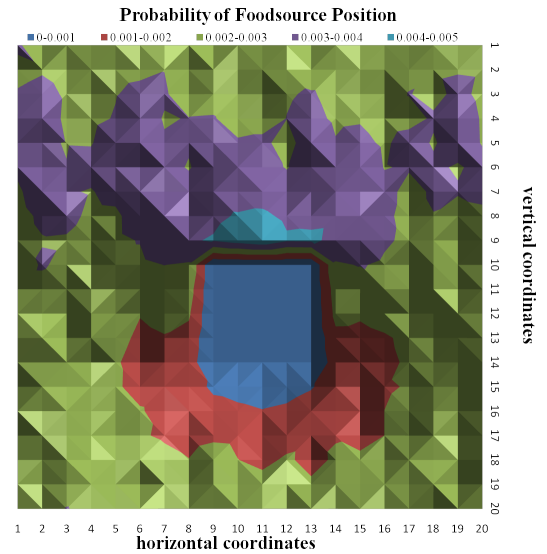


Figure 1: Graph showing  $P(F|A = north)$ , the probability distribution of  $F$ , the food source position, given a specific agent movement (in this case north). The data was obtained from 10000 single agent simulations in a  $20 \times 20$  grid world, agent position is (11, 11). Note that there is a peak north of the agent, meaning that it is more likely for the food source to be directly north of the agent when it moves north.

in its actions, and this is the case even if it does not have an explicit intention to communicate. This *digested information*, as discussed in [16], has several properties which are interesting for an observing agent with similar goals:

1. Actions *must* contain relevant information, even if the agent does not want to communicate
2. Better agent performance requires more, or the same amount of relevant information
3. The actions of an agent are likely to exhibit a higher density of relevant information than other parts of the environment
4. The actions of an agent might contain information that is not available in the current space or time.

From these properties it follows that a reasonable next step in our information maximisation model would be for the agent to use this digested information and incorporate it in their internal probability distribution. We extend the model so the agent can now, for all cells in its sensor range, detect whether one or more agents are in that cell and where they came from. So, the four new sensor states for each cell are agent that moved in from the north, ... south, ... east, ... west. Each observed move will lead to an adjustment of the assumed internal probability distribution, using a similar

form of Bayesian update already used to integrate the information from the environment. This adjustment of probabilities can be comfortably integrated into our existing infotaxis search.

Note that for the now described simulation all agents are equipped with those new “social” abilities and all of them use the other agents’ actions to update their internal world models. But they only use this ability if they accidentally encounter another agent. They do not deliberately seek out other agents.

## Bayesian Update

Let  $F$  denote the agent’s current internal probability model for the location of the food source  $F^*$ , and  $a$  the state of the random variable  $A$  that encodes the last move action of another agent it’s observing. The agent then use Bayes’ Theorem to update the probability distribution of  $F$ , with the observed action  $a$ .

What the agent is interested in is the probability of the food source to be in a specific location, given the evidence of another agent’s action and relative position  $P(F^* = w|A = a)$ . According to Bayes’ Theorem this is calculated for every potential location  $f$  of the environment as:

$$P(F = f|A = a) = \frac{P(A = a|F^* = f)}{P(A = a)} \cdot P(F = f) \quad (8)$$

Whenever an agent encounters one or several agents it uses this formula to adjust its internal probability  $P(F = f)$  for every location of  $f \in \mathcal{W}$ .

- $P(F = f)$ , the *a priori* probability, is the internal model of the agent for mapping the probability distribution of  $F^*$ , as gained by their own experience so far;
- $P(A = a)$  is the probability of an agent taking the move action  $a$ . Rotational symmetry suggests a probability of 1/4 for each action  $a \in \{north, west, south, east\}$ . Measurements in our single agent simulation confirm this. This is a normalisation factor, so the overall sum of probabilities is still one.
- $P(A = a|F^* = f)$  is the probability of another agent performing action  $a$  if the food is in position  $f$ . Note that the position  $f$  in this case will always be calculated in relation to the position of the observed agent. So, the question we are asking is for example “If the food is known to be 3 cells north of the agent, what is the probability of the agent performing move action  $a$ ”. We then record all the cases in the past where an agent has been observed 3 cells south of a food source together with the action it took.

To obtain these statistics for the computer simulation, we observed 10000 single infotaxis agents searching for the food. Note that the agents we used were non-social and thus

“blind” to the actions of other agents. They behaved according to the “Infotaxis” part of this paper. So, even though all the agents in the infotaxis simulation have the ability to sense other agents and update their internal world models, they still calculate their Bayesian update under the assumption that all others were non-social agents. We used the data obtained from non-social agents to create the statistics for the probabilities  $P(A = a)$  and  $P(A = a|F^* = f)$ .

After the agent updates  $F$ , it resumes the previously described infotaxis behaviour to generate its next move action. Note that agents which have successfully located the food stopped moving and were neither perceivable by other agents, nor blocking them. This was done to increase the challenge since it would have been trivial for another agent to infer from seeing another non-moving agent that the food must be within sensor range of that agent. As a result, the agents could not “cheat” by observing any agents which already knew where the food was.

This model, which includes the Bayesian update not only based on environmental variables, but also on other agents they encounter will be called the *Social Bayesian* model. Apart from the update of the internal model before the next infotaxis action is chosen, it is identical to the infotaxis model.

## Measurements

While flocking behaviour might be intuitively visible at this point in our model, defining an objective overall measure which quantifies the emergent flocking behaviour seems difficult. Instead, we aimed to measure the immediate effects of behaving according to the boids rules should have. We defined the following three measurements:

### Alignment

To quantify the *alignment* of the different agents, we added up all the agents’ movements and took the length of the resulting vector and normalised it. I.e., every agent  $x \in \mathcal{X}$  has an associated vector

$$\vec{v}_x \in \{(1, 0), (0, 1), (-1, 0), (0, -1)\} \quad (9)$$

corresponding to the last direction it moved in. The global alignment is then calculated as the length of the sum of all agent’s vectors, divided by the number of agents:

$$alignment = \frac{|\sum_{x \in \mathcal{X}} \vec{v}_x|}{|\mathcal{X}|} \quad (10)$$

This results in a value between 1.0 and 0.0. The maximum value is reached when all agents move in the same direction, and the lowest value of 0.0 is attained when the movement of all agents is distributed evenly between those moving north and south, and those moving west and east, respectively. Note again that agents which have found the food are not taken into consideration for this measurement, since

it would be irrelevant to measure how well aligned they are, once they are not moving anywhere.

This measurement are taken for every simulation step, and an average over all simulation steps is calculated for the whole simulation.

### Cohesion

To measure *cohesion*, we simply count, for every agent, how many other agents are within the agent's sensor range for any given time step. This value is then averaged over all agents, and over all time steps, and the result we call the local agent density, or simply density. This value, different from the global alignment, is only taken locally, and reflects how well agents keep other agents within their own sensor range.

### Separation

The hardest value to measure is *separation*, since it basically quantifies an objective of what should not happen. To approximate this, we measure how often one agent tries to enter the cell of another agent, and thus colliding with it. In this case, the agent trying to move will simply fail doing so. The resulting number of overall collisions is then divided by the number of time steps, providing an average amount of collisions per round, or simply collisions. This number is of course also dependent on the number of agents in the simulation, but this correlation if not linear, is therefore not normalised with respect to agent number. Thus, one needs to take care to only compare values where similar amounts of agents have been involved. Again, agents who have found the food are not considered for collisions detection.

### Results

All measurements were taken in a open ended simulation where the food had a 3 % chance of being moved every time step. When this happens, all agents' internal models are reset, and those agents who have already found the food are put back into the simulation. The simulations were run for 100,000 time steps, with 20 agents, in a  $20 \times 20$  torus-shaped grid world. As a baseline for comparison, we also measured those values for a group of agents that chose their actions at random, only stopping if they chanced upon the food source.

	Alignment	Density	Collisions
Random	0.23	1.03	0.72
Infotaxis	0.29	1.33	1.31
Social B. Update	0.39	1.68	0.49

Table 1: Flocking indication measurements taken for three behaviour models. (Random, Infotaxis, Social Bayesian

If we move from the random behaviour to the single agent infotaxis search, we see both the local agent density and the number of collisions increase. Since agents are not yet reacting to each other in the plain infotaxis model, this seems to

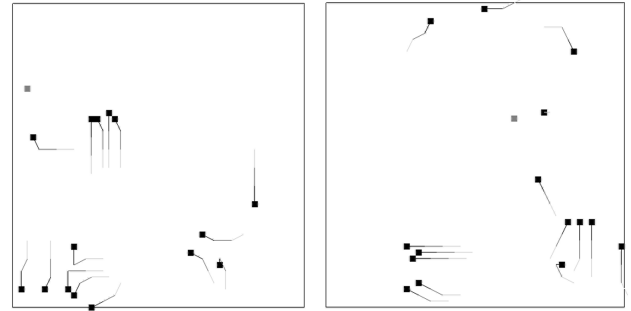


Figure 2: Two screen shots from a social infotaxis simulation with 15 agents, sensor range 5 in a 50 x 50 world. The grey box is the food source, the black boxes are agents. The lines indicate the vector of movement in the last 9 turns, in steps of 3.

be a result of the improved search algorithm. If we measure how long it takes, on average, for a *random* agent to find the food (ca. 450 time steps), and compare it to the time it takes an *infotaxis* agent to find the food (ca. 70 time steps), we see that the infotaxis search has a much better performance, resulting in agents actually finding the food before it changes position. This in turn leads to a local concentration of agents, which is likely to result in increased density and collisions. Note, however, that if we look at the alignment indicator we also see, that even for a group of agents that moves at random the average alignment is not 0.0, but 0.23. This is a statistical effect and not surprising, since it would actually take coordination to ensure that all agent's movements are always balanced between the different directions.

The interesting comparison is now between the two simpler models and the Social Bayesian Update. In the latter, we see a further increase in alignment, indicating that a high number of agents now move in similar directions during most of the simulation. Keep in mind that to achieve an average of 1.0, all agents would have to move in that same direction, in every turn. We also get a further increase in local agent density, while at the same time the number of collisions is reduced. So while there are now even more agents within the sensor range of each other, the agents manage to collide much less.

### Interpretation

We presented a model where the agents' behaviour is motivated by one single principle or goal, namely to gain as much information about a relevant variable in the environment. To achieve this, the agents take any kind of sensor variable, be it an environmental variable, such as the state of a grid world cell, or the action variables of another agent, and performs a naive Bayesian update on its internal proba-

bilistic model about said relevant variable. The agent's own actions are chosen in regard to which of them provides the greatest expected reduction of entropy, based on the agents' own internal model.

In this section, we would now like to discuss possible explanations on how this information maximisation model leads to the three different rules which create the boids-like flocking behaviour.

### Alignment

When an agent is controlled by non-social infotaxis behaviour moves, then its action contains information about the relative position of the food source. If we take a look at an agent moving north (due to rotational symmetry, the actual direction is exchangeable), then the food is more likely to be in a position north of the agent, and less likely to be in a position south of it. This effect, even though the agent does not know where the food is, results from the fact that the agent knows where the food is *not*. As seen in Fig. 2, the probability distribution has its highest peak directly north of the agent, and the minimum of the distribution is in the area south of the agent. Both peaks flatten out the further the cells are away from the agent.

Another agent who observed the first agent move north would perform a Bayesian update on its own assumed probability distribution of the food source. Everything else being equal, this would lead him to "believe" that the food is more likely to be north. The resulting move action would also be to rather move north than in any other direction. A flock of agents, each observing each other, could thereby create a "travelling wave" of high probability immediately outside of their sensor range, driving them all in a similar direction.

The generalised principle here is that an agent 1 observing actions by an agent 2 assumed to have similar goals would lead the original agent 1 to conclude that agent 2 has information that would make such an action reasonable, and in turn, this would make the same action more reasonable for agent 1.

### Separation

Whenever agent 1 observes an agent 2 moving in our grid world model, it performs a Bayesian update for the position of the food source. The biggest impact of this update is on the probabilities of the area immediately around agent 2. The cells of the world agent 2 observed in its previous turn are definitely empty, so most of the current area around agent 2 cannot contain any new information for the observer. So while observing another agent is an efficient way to gain information, the immediate environment around that agent becomes informationally unrewarding afterwards. An information-driven search would therefore try to steer away from the immediate area around an observed agent.

In general, if an agent 2 in a specific position reveals information it gets from being in that position to agent 1, then

the more information agent 1 gets from that agent, the less informationally interesting does being in that position become.

### Cohesion

In our current model, most of the cohesion seen in our agent groups seems to be a direct result of the high amount of agent alignment. If agents that meet each other move into a similar direction, with similar speed, then they also happen to stay together. In general, it would actually be reasonable to include a further term into the infotaxis mechanism which would account for the amount of information gained from other agents. Following from the "digested information" principle, it is informationally advantageous to keep other agents in sensor range, to be able to use them for a Social Bayesian Update. Seeing another agent, and being able to use the information in its actions increases each agent's expected entropy reduction.

All in all, if we take into account both separation and cohesion, the best solution in terms of information gain seems to be to keep other agents just inside your own maximum sensor range.

### Future Work

Since all agents observe each other we suspect there is the distinct possibility that a positive feedback loop can emerge, which detaches itself completely from the environmental information. As an example, an agent might take, for lack of better information, a random action; for example to move up north. Another agent might observe the first, and if it did not know anything apart from the fact that another agent moved north, he also would move north. The first agent in turn might now see the second, observe that the other agent moved north, and take this as good reason to also move north. This vicious feedback circle then continues, reaffirming both agents internal beliefs that "they are doing the reasonable thing". This phenomenon warrants further study, since it could illuminate how in social settings seemingly reasonable assumptions lead to strong "convictions" that are utterly wrong and detached from reality.

Furthermore, it might also be interesting to move the present model from a grid world scenario into a continuous world. This would not only create more realistic animations, but would also be necessary to establish that the observed effects are not just artefacts of the grid world model. The challenge here would be the extension of previously described information theoretic tools to the continuous domain.

### Conclusion

We found that information-based social observation mechanisms are able to reproduce several postulated mechanisms of flocking. This is confirmed both by qualitative observation as well as using quantitative measures. Starting with the assumption that every agent needs to obtain some kind

of relevant information from the environment to act intelligently, then most of the arguments follow directly from that. Infotaxis seems to be not only conceptually grounded, but both biological plausible ([25]), as it leads to behaviour that is very similar to actual moth behaviour, and reasonably efficient for some scenarios; its performance in these scenarios is close to that of an optimal strategy ([16]). Our extension to also include the information offered by other agent's actions is well motivated by the properties of "digested information", and the result is a performance increase beyond the level achievable for a single lone agent ([16]).

At this point in the argument, we already observe emergent flocking behaviour, only motivated by one single utility, the maximum information gain. Note that the relevant information we have been discussing does not necessarily have to be the location of a food source. It could refer to the position of predators, or the location of mates or other types of desirable states, and might lead to similar flocking behaviour via similar mechanisms. The *relevant information* hypothesis can also be applied to a wide variety of agent types, whether birds, fish, herd animals or humans, and could offer a possible *ab initio* explanation for an immediate evolutionary gradient leading to flocking behaviour for a diverse spectrum of organisms.

## References

- Ay, N., Bertschinger, N., Der, R., Gtler, F., and Olbrich, E. (2008). Predictive information and explorative behavior of autonomous robots. *European Journal of Physics B*, 63(3):329–339.
- Ay, N. and Polani, D. (2008). Information flows in causal networks. *Advances in Complex Systems*, 11(1):17–41.
- Capdepuy, P., Polani, D., and Nehaniv, C. L. (2007). Maximization of potential information flow as a universal utility for collective behaviour. In *Proceedings of the First IEEE Symposium on Artificial Life*, pages 207–213.
- Couzin, I. D., Krause, J., Franks, N. R., and Levin, S. A. (2005). Effective Leadership and Decision-Making in Animal Groups on the Move. *Nature*, 433(7025):513–516.
- Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley-Interscience, 99th edition.
- Crutchfield, J. P. and Young, K. (1989). Inferring statistical complexity. *Phys. Rev. Lett.*, 63(2):105–108.
- Der, R., Steinmetz, U., and Pasemann, F. (1999). Homeokinesis — A new principle to back up evolution with learning. In *Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA'99)*, Vienna, 17-19 February 1999.
- Dyer, J., Ioannou, C., Morrell, L., Croft, D., Couzin, I., Waters, D., and Krause, J. (2008). Consensus Decision Making in Human Crowds. *Animal Behaviour*, 75(2):461–470.
- Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2005). Empowerment: a universal agent-centric measure of control. In *Congress on Evolutionary Computation*, pages 128–135.
- Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2007). Representations of space and time in the maximization of information flow in the perception-action loop. *Neural Comput.*, 19:2387–2432.
- Laughlin, S. B. (2001). Energy as a constraint on the coding and processing of sensory information. *Current Opinion in Neurobiology*, 11(4):475 – 480.
- Pearl, J. (2000). *Causality: Models, Reasoning and Inference*. Cambridge University Press.
- Polani, D., Nehaniv, C. L., Martinetz, T., and Kim, J. T. (2006). Relevant information in optimized persistence vs. progeny strategies. In *Artificial Life X : Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 337–343. The MIT Press (Bradford Books).
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34.
- Rieke, F., Warland, D., de Ruyter van Steveninck, R., and Bialek, W. (1999). *Spikes*. A Bradford Book. MIT Press.
- Salge, C. and Polani, D. (2011). Digested information as an information theoretic motivation for social interaction. *Journal of Artificial Societies and Social Simulation*, 14(1):5.
- Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423.
- Sporns, O. and Lungarella, M. (2006). Evolving coordinated behavior by maximizing information structure. In *Artificial Life X : Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 323–329. International Society for Artificial Life, The MIT Press (Bradford Books).
- Tishby, N., Pereira, F. C., and Bialek, W. (1999). The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, pages 368–377.
- Todorov, E. and Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nature neuroscience*, 5:11.
- Tononi, G., Sporns, O., and Edelman, G. M. (1994). A Measure for Brain Complexity: Relating Functional Segregation and Integration in the Nervous System. *Proceedings of the National Academy of Science*, 91:5033–5037.
- Touchette, H. and Lloyd, S. (2000). Information-theoretic limits of control. *Phys. Rev. Lett.*, 84(chao-dyn/9905039. 6):1156.
- Touchette, H. and Lloyd, S. (2004). Information-theoretic approach to the study of control systems. *Physica A: Statistical Mechanics and its Applications*, 331(1-2):140 – 172.
- van Dijk, S. G., Polani, D., and Nehaniv, C. L. (2010). What do You Want to do Today? Relevant-Information Bookkeeping in Goal-Oriented Behaviour. In *Artificial Life XII: The 12th International Conference on the Synthesis and Simulation of Living Systems*, pages 176–183.
- Vergassola, M., Villermaux, E., and Shraiman, B. I. (2007). 'infotaxis' as a strategy for searching without gradients. *Nature*, 445(7126):406–409.

# An Innovative Bio-inspired Fault Tolerant Unitronics Architecture

M. Samie<sup>1</sup>, G. Dragffy<sup>1</sup>, Tony Pipe<sup>1</sup> and P. Bremner<sup>1</sup>

<sup>1</sup>Bristol Robotics Laboratory, Bristol, UK  
Mohammad2.samie@uwe.ac.uk

## Abstract

This paper presents the first implementation results of a novel Unitronics (Unicellular Electronics) architecture based system that uses a bio-inspired prokaryotic model. It is a programmable cellular FPGA-like system inspired by unicellular bacterial organisms, and transposes self-healing and fault tolerant properties of nature to electronics systems. An e-puck object avoidance robot controller was built to demonstrate all the underlying theories of our research, the validity of the bio-inspired model and the capabilities of the Unitronics architecture that it facilitated. The robot successfully demonstrated that it was able to cope with multiple, simultaneously occurring faults. Integrity of the system is continuously monitored on-line, and if a fault is detected its location is automatically identified. Detection will trigger a self-repair mechanism and only when it is complete will normal system operation resume.

## Introduction

Bio-inspired system design is a relatively new emerging field for the realisation of electronic systems. It attempts to learn from processes and characteristics of living things, such as self-replication and self-repair properties, adapting them to electronic systems. Bio-inspired systems depending on this type of motivation can be classified in two categories: Eukaryotics (multicellular) or Prokaryotics (unicellular) systems.

The early 90's saw the first attempts [1, 2] to construct bio-inspired electronics systems using a cellular array type architecture. They were based on properties and characteristics of and used mechanisms found in multi-cellular eukaryotic organisms. Here, similar to nature, all the cells of the system, in order to configure them for a specific function, contained a full or a partial copy of the organism's DNA (genome). This approach has invariably resulted in a large amount of DNA memory in each cell. The task of the memory is to store the genetic behaviour (DNA) of each cell of the system, in the form of configuration bits (genes) for both its functional characteristic and for the necessary interconnects. Embryonics and the POETic projects are examples of eukaryotic bio-inspired systems [3, 4]. CellMatrix offers an alternative approach for cellular implementation of systems [5].

Self-healing properties, immunological protection and learning abilities are amongst the advantages offered by the eukaryotic model. All previously proposed Embrionic systems suffer from several disadvantages:

- Inefficient functionality vs. silicon area requirement due to

large genome redundancy.

- Storing large amount of redundant information (each cell required a copy the entire DNA of the system or a large part of it) increases the probability of hardware faults and information mutation in the memory cells.
- Inefficient self-repair: row or column elimination kills an unnecessarily large number of healthy cells in response to the occurrence of a single fault.
- Demanding routing resources, especially for long-distance communication.

We suggest that if a model with at least similar performance advantages but based on a simpler form of biological life could be developed, then there is a chance that it might provide a solution to the above problems. We believe that the Unitronic artificial system, which is inspired by primitive unicellular beings called prokaryotes, in particular, bacteria, with its structure and characteristics does indeed offer the answer. It combats the problem of high genome redundancy, thus increases system reliability and is in all respect superior to all Embrionics based systems.

The novel artificial prokaryotic model we have proposed [6, 7] is a solution to build efficient fault tolerant hardware systems. It offers: efficient optimisation of genome redundancy, smaller silicon area, smaller memory for the storage of redundant (back-up) configuration information and requiring less logic support [6]. In our prokaryote model, the cell is only required to store its own configuration bits and some non-configuration bits that support self-repair and not a large part or the entire DNA of the system. Self-repair is achieved by a simple cell elimination process. A new self-test methodology was proposed [8] that offers an acceptable overhead compromise between time and hardware redundancy and guarantees that not only functionality, but all interconnect lines of the cellular system, are also tested.

## Prokaryotic Bio-inspired Model

The prokaryotic bio-inspired model is described in details in [6, 7] with a recommended self-test method given in [8]. This section summarises the main features of the model and the proposed self-test.

The prokaryotic bio-inspired model offers a multi-layer architecture of programmable universal cells. Each cell consists of a function unit (FU), a communication block and a memory block. The latter contains the configuration bits (gene) of the cell that define the required behaviour of both the function unit and that of the communication block, and non-



configuration bits which assist self-repair if a fault is detected. Since the task of the gene in the configuration register (CR) is to code the behaviour of a cell so it is termed as a *coding* gene, while the gene in the non-configuration register (non-CR) that assists self-repair is a *non-coding* gene. Thus each cell's genome could be viewed as consisting of one coding and one non-coding gene. The non-coding genes are assisting the functionality and the recovery of the coding genes both for the cell in which they reside and for other cells.

In a multi-layered prokaryotic model, cells form clusters, which in turn form colonies and on the top level biofilm communities are formed by colonies. Although the individual bacterial cells' genomes, in a family of species, are the same, due to continual evolution that takes place, mutation will differentiate them. Disregarding these small amounts of differences there will always be a strand in their DNA which they all share and is common to them all. Similarly therefore, in an artificial system family, clusters could be formed with cells that demonstrate similarity in their configuration bits. These cells, although they are unique and different in their own rights, do display similarity through a shared value (C<sub>sv</sub>) that is common to every cell in a cluster. Characteristics of artificial cells are stored in the form of bits in their configuration register and form its configuration vector (C<sub>cv</sub>). Therefore every cells' configuration vector is made up of a value that the cells share (C<sub>sv</sub>) and is common to them all, and by a differential value ( $\Delta g$ ) that distinguishes the cells from another. The configuration vector of a cell can therefore be described by Equation 1.

$$C_{cv} = C_{sv} + \Delta g \quad (1)$$

or generally as:  
 $C_{cv} = f(C_{sv}, \Delta g)$

where  $f$  refers to the evolutionary function and in the simplest form could be considered as XOR or subtraction functions.

Cluster forms the first community layer. It is a convenient collection of cells to aid self-repair. A cluster is a community of genetically related entities that need not have any functional relationship. In the simplest form, two different types of clusters may be defined: as shared value cluster (sv-cluster), and gene difference value cluster ( $\Delta g$ -cluster). The first one refers to those cells in the colony that have the same shared value of their configuration bits and hence originate from the same species. The second one refers to those cells that have the same genetic difference from their base species. Components of cells and clusters are shown in Fig. 1.

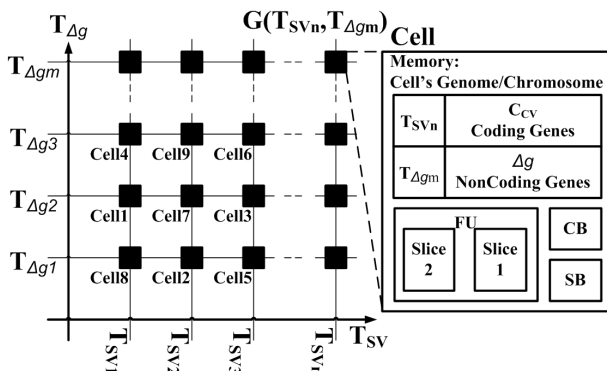


Fig. 1. A colony made up of inter-related clusters and cells

A colony layer is obtained where a correlation between different clusters exists. Colonies are groups of correlated cells that facilitate self-repair. Similarly to clusters they are genetically and not functionally grouped hardware entities. Our artificial colonial layer is equivalent with the biological mixed bacterial colony and is made up of several sv-clusters (species). When a new daughter cell for one of its species is created the species shared value is differentiated by  $\Delta g$ . This differentiation process in nature amongst different bacteria occurs through the horizontal gene transfer mechanism (HGT). Here genes are transferred from one bacterium to another that changes their characteristics (e.g. acquire antibiotic resistance). HGT, in an artificial system, provides a correlation mechanism between different sv-clusters, so that  $\Delta g$  of a cell in one sv-cluster can be used to evolve the gene of another cell in another sv-cluster. In this case the shared value of the new cell is differentiated with the  $\Delta g$  from another cell, Fig. 2.

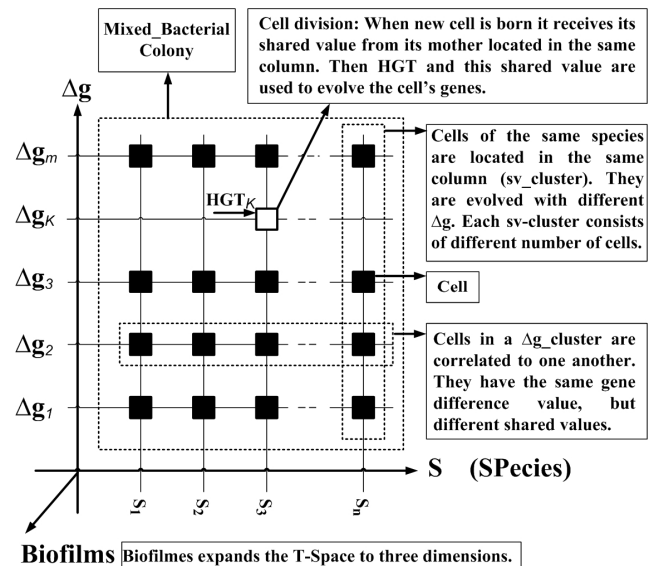


Fig. 2. Prokaryotic Bio-Inspired Model.

## T-Space

Let's suppose that an artificial system, as shown in Fig. 3a, consists of  $x$  number of cells, where  $x = n \cdot m$  and the configuration vectors of the cells are  $C_{cv1}, C_{cv2}, \dots, C_{cvx}$ . In this case the genome of the system ( $G$ ) could be described by a set of genes of the individual cells as:

$$G^p = \{g_1, g_2, \dots, g_x\} \\ = \{C_{cv1}, C_{cv2}, \dots, C_{cv(m,n)}\} \quad (2)$$

where  $g$  stands only for the configuration vector ( $C_{cv}$ ) part of the cell's memory and excludes the non-configuration bits. In system's genome  $G^p$  also shows how this  $x$  set in the physical space is defined by  $T_{sv}$  and  $T_{\Delta g}$  addresses.

If we now also include the non-coding genes (non-configuration vector) of the cells in their genome  $G$ , then the HGT (horizontal gene transfer function) function will map the coding genes from physical space (equation 2, Fig. 3a) to a new set of two dimensional T-Space (Fig. 3b), that is defined

by Tsv and T<sub>Δg</sub> address tags.

$$G^T = HGT(G^P) \quad (3)$$

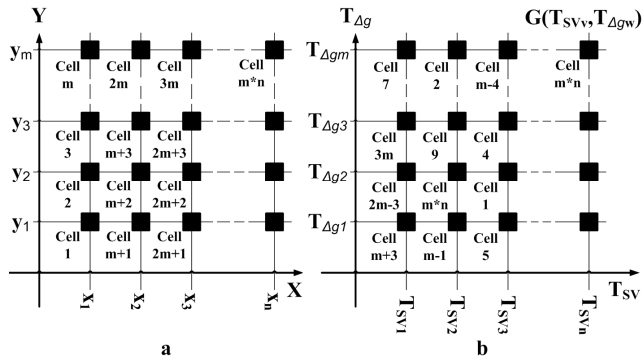


Fig. 3. Example of cells' placement : a) physical, b) T-Space.

With this HGT function, inspired by bacterial communities and differences in its species, artificial cells in clusters can also be defined by a common strand and their differences. Thus grouping of cells into sv-clusters and  $\Delta g$ -clusters will show their similarities and differences, which are also identified by the Tsv and T<sub>Δg</sub> address tags. If tag combinations are unique, then to refer to any specific and individual cell in the array, instead of physical addresses tags could be used. The HGT function will transfer the gene of the  $i^{\text{th}}$  cell of array, addressed by  $i$ , from the physical space into tag space as:

$$g(Tsv, T_{\Delta g}) = HGT(\text{Cell}^i) \quad (4)$$

where  $g$  is the configuration vector (Ccv), the coding gene of the cell. The Tsv shared value tag (Csv) identifies a group of similar cells. The T<sub>Δg</sub> differential parameters tag refers to a group of cells that have already been differentiated with the same  $\Delta g$  that  $i^{\text{th}}$  cell needs to be evolved with. Therefore equation 4 could be rewritten as:

$$\begin{aligned} Ccv_{(Tsv, T_{\Delta g})} &= f(Csv_{(Tsv)}, \Delta g_{(T_{\Delta g})}) \\ Tsv &= \{1, 2, \dots, v\} \\ T_{\Delta g} &= \{1, 2, \dots, w\} \end{aligned} \quad (5)$$

Where  $v$  is the number of shared values and  $w$  is the number of differential parameters (gene differences).  $v$  could also be considered as the number of different species of cells which collectively define the system. Function  $f$  in equation 5 could be any simple logical or algebraic function such as XOR, summation or subtraction of the shared value and the differential parameter. This equation precisely describes the functionality of every cell during its normal, test and self-repair modes of operation using a configuration vector (Ccv), a shared value (Csv) and a differential parameter ( $\Delta g$ ). Tsv and T<sub>Δg</sub> tags together assign a unique address to every cell. This address is only a 'soft' entity and is not used as a sequential physical address location of cell placements in the unitronic architecture. Instead cells based on their tag addresses are grouped to achieve the best possible compression and correlation solution for clusters and colonies. The number of cells in the array is always  $x = n \cdot m$ , where  $n$  and  $m$  may have different values to  $v$  and  $w$ . This means that tag addresses do

not refer to a physical cell because such cells do not actually exist in the array.

Shared value (Csv) given in equation 4, is a non-existent entity and there are no cells in the unitronics array that include such value in their memory. It is the result of a compression operation and a feature of the bio-inspired prokaryotic model. Genome of the cell ( $G$ ) can be defined as:

$$\begin{aligned} G(Tsv, T_{\Delta g}) &= \{g(\text{coding}), g(\text{non-coding})\} \\ &= \{(Ccv_{(Tsv, T_{\Delta g})}), (Tsv, T_{\Delta g}, \Delta g)\} \end{aligned} \quad (6)$$

Biofilms are the top layer of bio-inspired prokaryotic model. This is another software entity that expands T-space from 2 to three dimensions. Here colonies are grouped so that a faulty cell in one colony may be correlated to other cells in other colonies. In this case, to facilitate the repair of faulty cells, a larger search area is available in the T-Space world.

### Self-Repair

Although each and every cell has its own BIT (Built-in-Self-test), colony is the lowest level that supports system self-repair. Functional system operation is synthesised to cell and not community level (cluster, colony, and biofilms). Each cell in the array, through its individual configuration vector (Ccv), is programmed to do a specific task so that the cells collectively execute the required functionality and define overall system operation. If faulty operation is detected community layers will provide system recovery self-repair support.

For the sake of the foregoing discussion let us consider the system's genome, consisting of Ccv,  $\Delta g$ , Tsv and T<sub>Δg</sub>, as a software entity, and all the functional, communication elements of the cells and their physical memory requirement for genome storage, as hardware entities. Faults may develop in both the software and in the hardware part of system. If the fault is hardware related then its associated cell will need to be killed and operationally eliminated from the system. In this case through the process of cell division a new cell, of the same species (same Csv) as the faulty one, should be 'given birth' during which, to recover the system, a repair process will take place.

Cell division requires a 'new' cell which during the repair process will be configured the same as the eliminated faulty cell. Since, unlike in nature, our current technology does not facilitate birth of hardware cells, artificial systems must have some redundancy through the availability of spare cells. If a system consists of  $n$  available cells of which a specific application uses  $m$  cells, then the number of available spare cells is  $n-m$ .

Consider that cell  $k$  (between cells 1 to  $m$ ) is detected as faulty (Fig. 4). In this case all cells located between  $k+1$  to  $m$  are shifted one cell forward to cells  $k+2$  to  $m+1$ , where cell  $m+1$  is part of the system's redundant available cells. Cell  $k+1$  will act as a 'spare cell' and will replace the faulty cell. Cell division is a two step process:

- Shifting prepares a spare cell adjacent to the faulty one.
- Calculating and loading the shared value of faulty cell into the spare cell.

These will be followed by a differentiation process where from the shared value the cell's configuration vector (Ccv) will

be evolved.

Lack of the shifting process is the only difference between hardware and software fault repair. If several faulty cells simultaneously develop a fault then, following their elimination, the same shifting process will take place and the number of available redundant cells will be accordingly reduced. During shifting, cells are individually checked for integrity and simply by-passed if they were previously killed, while their neighbours will serve as spare cells and will take over the functionality of the faulty ones.

An example of a system consisting of  $n$  cells is shown in Fig. 4b. Here the implementation of a specific application requires  $m$  number of cells and the remaining ones are redundant cells acting as available spare cells. Fig. 4b shows the situation when two cells simultaneously develop a fault. The faulty cells (shown in black) are killed (Fig. 4c) and all cells are shifted to prepare a spare cells next to the faulty ones.

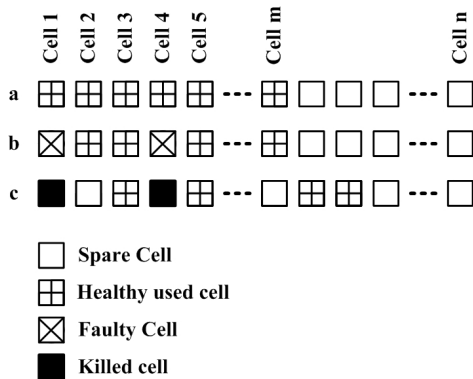


Fig. 4, Shifting process of self-repair mechanism.

We mentioned previously that clusters are communities of software related cells that have the same shared value, or the same differential parameter. The genome ( $C_{Gen}$ ) of a  $sv$ -cluster is made up as a union ( $\cup$ ) of the genes ( $g$ ) of its individual cells and can be expressed as:

$$C_{Gen}(T_{sv_i}) = \cup g(T_{sv_i}, T_{\Delta g_j}), \quad (7)$$

$$i \in \{1, 2, \dots, v\},$$

$$j \in \{1, 2, \dots, w\}$$

where  $j$  refers to the individual cells in the cluster having the same shared value addressed by  $T_{sv_i}$  and  $i$  refers to the  $i^{th}$   $sv$ -cluster,  $T_{sv_i}$ . These clusters are shown by the vertical lines in the Fig. 5. A similar equation can be formulated for  $\Delta g$ -clusters that have the same differential parameters:

$$C_{Gen}(T_{\Delta g_j}) = \cup g(T_{sv_i}, T_{\Delta g_j}), \quad (8)$$

$$i \in \{1, 2, \dots, v\},$$

$$j \in \{1, 2, \dots, w\}$$

where  $i$  refers to the individual cells in the cluster having the same differential parameters addressed by  $T_{\Delta g_j}$  and  $j$  refers to the  $j^{th}$   $\Delta g$ -cluster,  $T_{\Delta g_j}$ . These clusters are shown by the horizontal lines in Fig. 5. It also shows an example of how the physical placement of a faulty cell in the array differs from its placement in T-Space.

Every cell in Fig. 5 has its place both in the  $sv$ -cluster and

in the  $\Delta g$ -cluster. When faults are detected, for as long as one healthy cell exists in both  $C_{Gen}(T_{sv_i})$  and in  $C_{Gen}(T_{\Delta g_j})$ , the gene of faulty cell can always be recovered with  $T_{sv_i}$  and  $T_{\Delta g_j}$ . Fig. 5 also shows that cells do not need to be physically sorted when comparing their locations in T-Space.

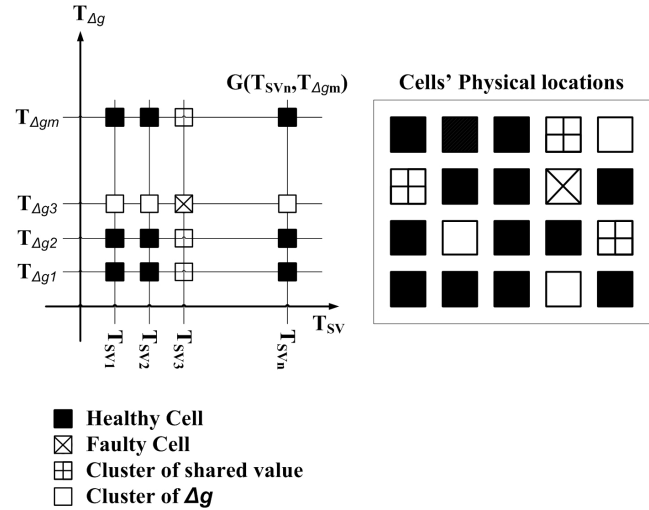


Fig. 5, An example of faulty cell, its physical placement in the array, and in the T-Space.

Equation 7 shows that how, in a prokaryotic model based system, clusters compress the system's genome. Every cell in the appropriate clusters of  $C_{Gen}(T_{sv})$  (vertically sorted in Fig. 5) is expressed with a same shared value and some differential parameters. The self-repair process uses this shared value during cell division by copying that of the faulty cell into the spare cell. It is only the differential parameter ( $\Delta g$ ) that distinguishes the cell now from other cells in the cluster. The healthy configuration vector can be recovered by differentiating this shared value with the faulty cell's  $\Delta g$ . It can be extracted from the  $\Delta g$ -cluster of  $C_{Gen}(T_{\Delta g})$  by  $T_{\Delta g}$ , where the faulty cell belonged. Since all cells in a  $sv$ -cluster have the same  $C_{sv}$ , it is readily available from any of its cells. It is a calculable entity and therefore requires no storage. Finally, the configuration vector of the faulty cell can be calculated as  $C_{cv_i} = C_{sv_i} + \Delta g_j$  (Equation 5). For safety and for easy self-repair purposes neither  $\Delta g$  nor  $T_{\Delta g}$  is saved in the cell's own non-configuration register but another cell will host them. In this way, every cell in the cluster has a back-up memory in the form of a non-configuration register that stores information for other cells.

Self-repair process takes place in three steps:

- Cell division.
- Identifying the species of the faulty cell, the  $sv$ -cluster and the actual shared value.
- Differentiating the shared value with  $\Delta g$  obtained from,  $\Delta g$ -cluster.

Steps 2 and 3 can only be executed if the faulty cell's tags remains healthy. Since the bit requirement of the tags is considerably less than that of  $C_{cv}$  and  $\Delta g$ , this condition is not difficult to meet. However should the tag values still mutate, additional safety storage is provided by fault tolerant RAMs in an external backup memory.

## Self-Test

The bio-inspired self-test we are proposing is based on two characteristics of biological systems:

- In nature, the DNA is a double helix, a duplicated sequence of complementary genes. It means that both sequences define exactly the same organism with exactly the same features. Therefore one strand is sufficient for the growth and development of an organism [9].
  - Transposons (formally termed *jumping genes*) are sequences of DNA that can move around to a different position within the genome of a single cell. Such mobile genetic elements can move within the genome from one position to another using a “cut and paste” mechanism [10].
- These two characteristics found in nature can be used to inspire the development of a bio-inspired self-test model for artificial systems by observing that:

- If we could guarantee that by configuring the processing elements of an artificial cell with both its gene and complementary gene, their functionality would remain the same and
- That using the concept of the *jumping genes* mechanism could offer a solution to switch over and substitute input signals of such processing elements and interchange their outputs.

The DNA is a double helix of two complementary genetic sequences. Both sequences will configure the cell for exactly the same function. Fig. 6 shows the placement of cells for the proposed artificial prokaryotic cell when the cell is configured by the sequence of the original genome and by its complementary (\*) one. Because of the nature of the sequences it is sufficient to store only one of them in the cell's memory.

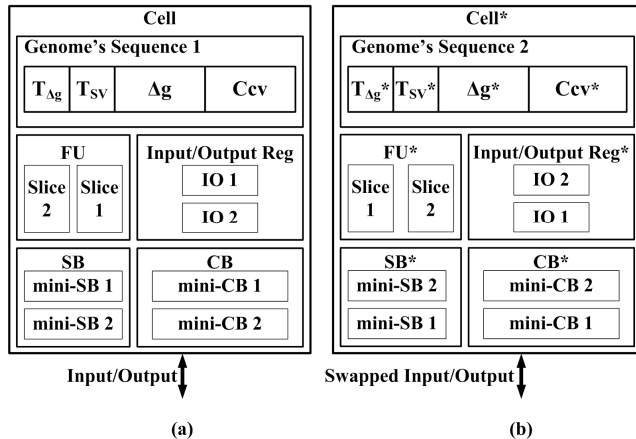


Fig. 6, A cell configured in two different modes, normal and test modes.

All functional components of the cell, such as FU, SB, CB and IO registers, are in pairs (Fig. 6). In normal operation they are cascaded to implement a higher order function. For instance, a SB is divided to two mini-SBs. Each mini-SB has a simple switching function, but joined together they can implement more powerful functions. If the controlling genes of mini-SB 1 and 2 are switched over, their functionality will also be switched over. Applying rules i. and ii. to Fig.6 a new test methodology is created. Configuration vector Ccv and its Ccv\*

complement will respectively configure the circuit for a normal (Fig. 6a) and a complementary (test) mode (Fig. 6b).

Cells of the array execute their assigned functions in one machine cycle. The cycle however is divided into four discrete sequential activities:

- Update of inputs.
- Normal mode of operation
- Switch over genes and switch over inputs and outputs.
- Test mode: check results
- Switch back genes, and inputs and outputs
- update outputs (cell passed) or kill cell (cell failed)

During a machine cycle both the functionality of the cells' components are switched over and also their external signals are swapped round. Only such simultaneous swap and switch mechanism can guarantee correct functional set-up and input data for self-test. Detailed description of this algorithm is given in [8].

In normal mode of operation all cell output results are saved but not yet propagated. In the following test mode all cells are subjected to input swap and functional change over. These results are also saved. If it is found that the two results correspond then their outputs are released and normal functional operation can continue. If however the outputs differ then self-repair is requested. Only once this is complete and error free operation is recover, will normal system operation resume.

## Unitronics Architecture

Embryonics, inspired by multi-cellular eukaryotic organisms, was the first project that attempted to map biological processes to electronic hardware. A newly emerging field that uses models of prokaryotic organisms such as bacteria to create bio-inspired man-made systems is a related but different architecture. Here, we name the artificial electronic systems inspired by these unicellular creatures, 'Unitronics' [6, 7, 8]. The Unitronics system uses two different types of cells; core cells (C-cell), surrounded by peripheral cells (P-cell) around its perimeter (Fig. 7). The basic architecture of both cell types is based on the block diagram of Fig. 6, except that P-cells do not have a function unit (FU).

Core cells are configured to implement specific functions, as defined by the genes in their configuration register. Peripheral cells on the other hand only manage the input and output information flow, including signal swapping during test mode. Unitronics adapts a 'see-of-gates' architecture (Fig. 7) similar to that used by commercial FPGAs but partitions the system into prokaryotic islands. Islands are formed by groups of C-cells surrounded by P-cells.

Peripheral cells (Fig. 8) of the array provide an interface between the island of C-cells and the outside world. They consist of two flip-flops and a signal controller. They have four bi-directional pins, two of which (P1 and P2) provide communication with the peripheral bus (P-BUS), and the other two (E1 and E2) provide communications with the global bus (G-BUS). Signal directions in E and P are defined by the appropriate configuration bits for the P-Cell. The flip-flops receive their data either from the External (E) or from the Peripheral (P) bus lines, under the control of two multiplexers. External communication can be disabled in order to swap data

of P1 and P2. This is accomplished by the two flip-flops; connected in this case as a circular shift register.

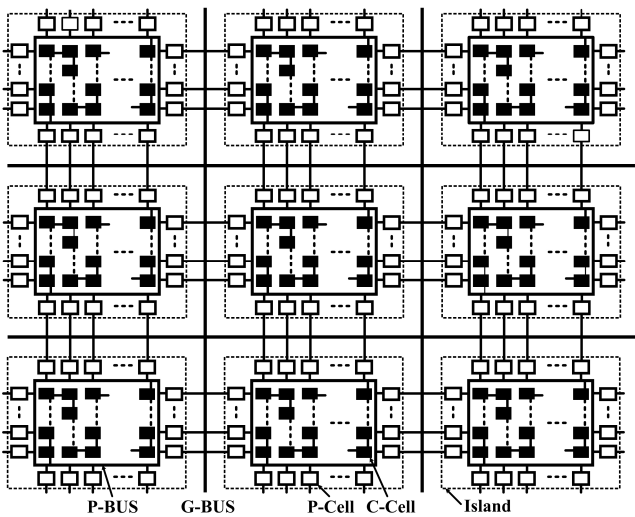


Fig. 7, Schematic diagram of Unitronics.

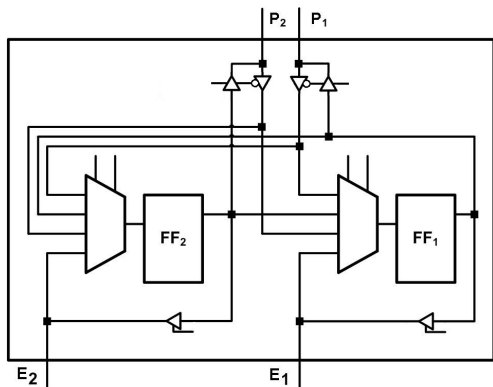


Fig. 8, Peripheral cell, P-Cell.

During test mode, data from the P-lines are loaded into the flip-flops are swapped round, and placed back onto the same lines. As a result the lines now have swapped data, as compared with what they had before. Fig. 8 shows only those components of the peripheral cell that provide data switching between P1 and P2 lines.

The array has 2 different types of buses: G-BUS, P-BUS and P-BUS (Peripheral Bus). G-BUS is used for distant communication between C-Cells in different islands via their own P-Cells where signal swapping is also possible.

P-Cells provide flexible connection between any two lines of the G-BUS to any two P-BUS lines. Lines are grouped in pairs, so that once a line is selected as input/output from G-BUS to P-BUS, the second line provides switch over when (e.g. in test mode) required. For self-repair there are additional redundant spare P-Cells.

P-BUS, on entering the array of C-Cells, is divided to C-BUS (Configurable Bus) and L-BUS (Local Bus). They are interconnecting wires, lines and channels, similar to commercial FPGAs. C-BUS provides the required cell to cell

interconnect. It is configured by the core cells according to their functional and communicational requirements. Lines of the configurable bus can be grouped, cut, joined and swapped. The bus also supports cell elimination during self-repair if a cell developed a hardware fault. In this case, the faulty cell is killed, its functionality is shifted to the next cell along the configurable bus and all preceding cells are also shifted until a healthy stand-by cell is found. The L-BUS, though can be divided to sub-sections, usually passes through the cells and only makes connection to those with which long distance data communication is required. It is local to the island, and would normally connect to the P-BUS only at the first and the last cell of the island.

C-Cells are the processing and communication elements of the system and as such they provide processing Function (F), signal Routing (R), information storage as Memory (M), and switching as Void (V) tasks. The two slices of the cell can work in tandem and undertake any combination of the above tasks as for instance FF, FR, MV, RM and etc. The detailed architecture of configurable bus is beyond the scope of this paper, but its important characteristics are indicated in Fig. 6. The cell's Connection Box (CB) manages how the cell should be connected to the network of other cells in the island. Inputs to the cell's Function Unit (FU) are provided either from the bus via the CB or from the cell's neighbours via dedicated neighbouring connections lines.

FU includes two 2-bit slices. Each slice is supported by the cell's genome, which is essentially an LUT. It can either define the precise function the slices should execute, or can configure them for signal routing. Slice function can either be logical or algebraic. When for example a cell is configured as RF then slice 2 will undertake signal routing, while slice 1 will execute a function on its output. FF set-up enables the cell for a more sophisticated function.

The cell can be used as a memory to implement registers, counters and, in case of a distributed memory, an 8, 16, 24 or 32-bit RAM. It is called a distributed memory because one cell can only provide upto two memory locations. The configuration bit (Ccv) register is not an addressable memory. To allow such functionality a distributed memory feature has been designed. In this case another cell is used as a memory controller. When the cell acts as a "Void" it provides a connection between C-BUS and L-BUS. If a cell is used for M or V the functionality of its slices' is reduced.

In summary the Unitronic architecture, inspired by biological colonies and the circulatory system of a Biofilm, is a network of colonies supported by adequate routing and communication facilities for the cellular array. Both hard and 'soft' entities of the architecture demonstrate biological inspiration. Cells, islands and the circulatory system are the hardware components, and clusters, colonies and biofilms are the 'software' components of the Unitronic system (Table 1). There is no physical location in the array that can be identified as being a cluster, or colony. Both are 'soft components' providing immune protection for the system for fault detection and repair. The architecture in Fig. 7 is a substrate where cells, cluster, colonies and biofilms are grown in the islands located in the network of voids and circulatory system.

	Software	Hardware
Cell (C-Cell, P-Cell)	Yes	Yes

Cluster	Yes	No
Colony	Yes	No
Biofilms	Yes	No
Island	No	Yes
Circulatory System	Yes	Yes

Table 1. Hard and 'soft' entities of Unitronics

## Robot Controller Demonstrator

In this example, to demonstrate the self-healing and self-repair capability of Unitronics, the timer part of a movement controller for an e-puck object avoidance robot (Fig. 9a) from EPFL [11] is implemented on a Unitronics array. The block diagram of the robot control system, operating in normal environmental conditions is shown in Fig. 10. The Unitronic timer part is synthesised on a Xilinx XUPV5-LX110T development board (Fig. 9b) [12], while the movement part of the controller and the interface between the robot and the Unitronics system is provided by Matlab. Using hardware co-simulation, data from the Unitronics array is transferred to Matlab in a 2-bit data. One bit defines whether a right or left turn is required from the robot, while the other is a fault indicator for the Unitronic system.

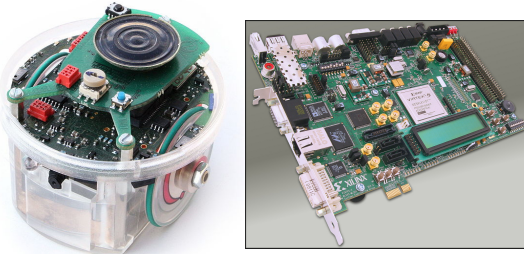


Fig. 9. a) e-puck, b) XUPV5-LX110T

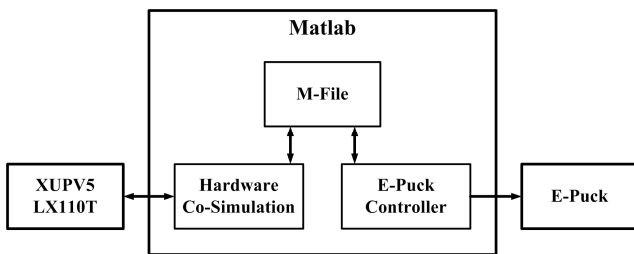


Fig. 10. Block diagram of the robot controller system

The timer is a 16-bit up counter the implementation of which required eight Unitronic cells. Fig. 11 shows the cells' genomes that implement the timer. The slices of all the cells, in this example, are configured as function-function (FF) and define a full adder. In reality the circuit offers a 16-bit full adder, but with inputs set to '0' and carry-in set to '1', it behaves as a 16-bit counter. MSB bit of this counter describes whether robot should turn right or left. Combination of turning

right and left makes the robot to move in a figure 8-like manner. Since the genome of every cell is the same, their identical Csv translates into one sv-cluster and their  $\Delta g$  (equalling to zero) into one  $\Delta g$ -cluster.  $T_{\Delta g}$ , and  $T_{cv}$  tag values are chosen arbitrarily as "10" and "11" respectively.

Since all cells are located in the same sv-cluster and in the same  $\Delta g$ -cluster, fault recovery is always guaranteed for as long as there is one healthy cell in the system. This example uses the simple algebraic function in Equation 9:

$$C_{cv}(T_{sv}, T_{\Delta g}) = C_{sv}(T_{sv}) + \Delta g(T_{\Delta g}) \quad (9)$$

Since in this example  $\Delta g = 0$  means that  $C_{cv} = C_{sv}$ . Consider a situation when seven out of the 8 cells are faulty and only one functions correctly. If we assume that all tags are correct and cell 5 is the faultless cell then after eliminating the faulty cells the next step is a shift process. With this, if the cells are sequentially placed along the bus, cell 1 will assume the position of cell 5 and the remaining cells occupy positions cell 9 to cell 15 of the stand-by cells.

Cell No.	Cell's FU		Cell's Genome			
	Slice 2	Slice 1	TAg	Tev	$\Delta g$	Ccv
Cell 8	FA	FA	10	11	0000000000000000	000340C00406969
Cell 7	FA	FA	10	11	0000000000000000	000340C00406969
Cell 6	FA	FA	10	11	0000000000000000	000340C00406969
Cell 5	FA	FA	10	11	0000000000000000	000340C00406969
Cell 4	FA	FA	10	11	0000000000000000	000340C00406969
Cell 3	FA	FA	10	11	0000000000000000	000340C00406969
Cell 2	FA	FA	10	11	0000000000000000	000340C00406969
Cell 1	FA	FA	10	11	0000000000000000	000340C00406969

Fig. 11. Unitronic timer implementation (values shown in hex)

The next step is to search in the sv-cluster space and identify the faulty cell's shared value. This is achieved by sending a token that will locate the first faulty cell, in this case cell 15. In order to find the shared value of this cell its  $T_{sv}$  tag is sent to all cells in the cluster. Since only cell 12 is healthy, the tag requests the extraction of its shared value using the rearranged form (i.e.  $C_{sv} = C_{cv} - \Delta g$ ) of equation 9. This here will coincidentally yield the same as the  $C_{cv}$  value of cell 12 and be released to the bus. All those cells which need the recovery of their shared value and have the same  $T_{sv}$  as cell 15, will receive it. In this case it will affect all cells of the cluster except cell 12. The final step of the repair process is to differentiate it with all the faulty cells'  $\Delta g$ . Since  $\Delta g$  is zero for them all, their configuration vector can now be simultaneously recovered, using equation 9.

In this example cluster identification is trivial due to the repetitive nature of the cell functions required. This in larger digital systems becomes more difficult. These however are typically composed of regular building blocks, i.e., registers, counters, multipliers etc; where this regularity can be exploited to simplify cluster formation. Our fault recovery mechanism is applicable to circuits with any complexity. Since motion cannot be demonstrated on paper the actual

behaviour with run-time fault detection and fault repair is shown under the following youtube link <http://www.youtube.com/watch?v=GO0YfVf0tMw>

Another example of a PD controller is shown in Fig. 12. The waveform illustrates the actual behaviour of the hardware (not simulation results!) and the fault recovery process of the controller. The PD controller was also implemented, also as an interim step before VLSI implementation, on a Xilinx XUPV5-LX110T development platform. The controller required 40 Unitronic cells and a ‘soft’ fault was injected in the genome of cell 3.

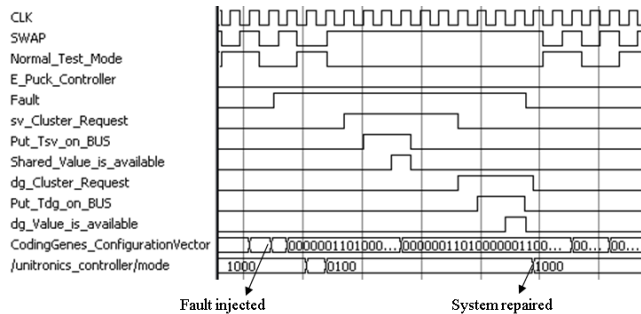


Fig. 12, Implemented robot controller fault recovery.

During the operation of the robot controller a fault was inserted into cell3. Fig. 12 shows the fault recovery process of the implemented system:

1. Fault is injected at *fault injected* point into the system.
2. The effect of the fault causes the gene to mutate at *CodingGenes\_ConfigurationVector*.
3. Simultaneously self-test using input data and control sequence complementation recognises it, identifies the faulty cell and initiates self-repair.
4. Self-repair requests the mutated faulty cell's  $C_{SV}$  at *sv\_Cluster\_Request*. For this  $T_{SV}$  at *Put\_Tsv\_on\_BUS* identifies the cluster and the cells that share the same portion of the configuration vector with the faulty cell. With the aid of the cluster's cells,  $C_{SV}$  is calculated at *Shared\_Value\_is\_available*.
5. Recalculation of the faulty cell's corrupted  $C_{CV}$  configuration vector also requires its  $\Delta g$ .
6.  $\Delta g$ 's address  $T_{\Delta g}$  is triggered at *Put\_dgTag\_on\_the\_BUS* in order to locate the same  $\Delta g$ .
7. When  $\Delta g$  is also available, using Equation (9) the faulty cell's  $C_{CV}$  can be calculated (*dg\_Value\_is\_available*='1').
8. With its recovery, on-line repair of the faulty cell is complete and the recovered correct response result of the cell is now allowed to propagate to its final output.
9. Normal system operation (at *System repaired*) in the next machine cycle resumes as if fault never occurred.

## Conclusion

On-line fault detection and fault repair capability of our Unitronics architecture, based on the bio-inspired prokaryotic model, is demonstrated using an e-puck object avoidance mobile robot. Implementation of the robot required 8 Unitronic cells appropriately interconnected and then mapped onto a Xilinx XUPV5-LX110T development board. The fault

tolerance model of the system guarantees that “if similarities and differences between healthy and faulty cells are known then, full recovery of any Unitronic implemented system is possible”. The system is able to cope with and repair any number of simultaneously occurring dynamic (SEU) or static (hardware) faults. The amount of fault repair only depends on the number of spare cells the system is equipped with. Its fault repair uses significantly less memory for gene storage and considerably less hardware overall for target system implementation than any previously proposed bio-inspired architecture.

In future work we plan to undertake a more detailed performance analysis as a function of the number of errors, investigate the implementation of more complex digital systems, and look at the implications for cluster formation. Additionally, we plan to investigate implementing higher level fault tolerance techniques using Unitronics as the substrate.

## Acknowledgement

This research work is supported by the Engineering and Physical Sciences Research Council of the United Kingdom under Grant Number EP/F062192/1.

## References

- [1] Garis, H. de. (1993). Evolvable Hardware. The Genetic Programming of Darwin Machines [C]. In *Proceeding of Artificial Neural Nets and Genetic Algorithms*, pages 441-449.
- [2] Mange, D. (1996). Embryonics: a new family of coarse-grained FPGA with self-repair and self-reproducing properties. Towards Evolvable Hardware: An evolutionary approach. *Springer Verlag*. Pages 197-220.
- [3] Mange, D. and Sipper, M. and et al. (2000). Towards Robust Integrated Circuits: The Embryonics Approach. *Proceedings of the IEEE*, vol.88, no.4, pages 516-541.
- [4] Barker, W., Halliday, D. M., Thoma, Y., and et al (2007). Fault Tolerance using Dynamic Reconfiguration on the POetic Tissue, *IEEE Transactions on Evolutionary Computation*, Vol. 11, No. 5, pages 666-684.
- [5] Macias, N. Durbeck, L. Prokopenko, M. (2008). *Advances in Applied Self-organizing Systems*. Springer.
- [6] Samie, M., Dragffy, G. Pipe, T. and et. Al (2009). Prokaryotic Bio-Inspired Model for Embryonics. *AHS'09 - NASA/ESA Conference on Adaptive Hardware and Systems*, pages 163-170.
- [7] Samie, M. Dragffy, G., Pipe, T. and et al (2009). Prokaryotic Bio-Inspired System. *AHS'09 - NASA/ESA Conference on Adaptive Hardware and Systems*, pages 171-178.
- [8] Samie, M., Dragffy, G., Pipe, T. (2010). Bio-Inspired Self-Test for Evolvable Fault Tolerant Hardware Systems. *AHS2010 - NASA/ESA Conference on Adaptive Hardware and Systems*. pages 325 – 332.
- [9] Jacob, F., Brenner, S., Cuzin, F. (1963). The Regulation of DNA Replication in Bacteria. *Cold Spring Harbor Symposia Quantitative Biology*. pages 329-348.
- [10] Kidwell, M. G. (2005). Transposable elements. In ed. T.R. Gregory. *The Evolution of the Genome*. San Diego: Elsevier. Pages 165-221. ISBN 0-12-301463-8
- [11] Mondada, F., Bonani, M., Raemy, X. and et al. (2009). The e-puck, a Robot Designed for Education in Engineering. *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, vol. 1, num. 1, pages 59-65.
- [12] XUPV5 - LX110T User manual, <http://www.xilinx.com/univ/xupv5-lx110T-manual.htm>



# A biomimetic robot controller based on minimizing the unpredictability of the environment: allostatic control revised

Martí Sánchez-Fibla<sup>1</sup>, Armin Duff<sup>1</sup>, Ulysses Bernardet<sup>1</sup> and Paul F.M.J. Verschure<sup>1,2</sup>

<sup>1</sup> SPECS, Technology Department, Universitat Pompeu Fabra, Carrer de Roc Boronat 138, 08018 Barcelona, Spain.

<sup>2</sup> ICREA, Institució Catalana de Recerca i Estudis Avançats, Passeig Lluís Companys 23, 08010 Barcelona.  
{marti.sanchez,armin.duff,ulysses.bernardet,paul.verschure}@upf.edu

## Abstract

Rodents are optimal real-world foragers that regulate internal states, such as security, arousal, energy, etc., maintaining a dynamic stability with their surroundings. Free exploration is an interesting scenario as rodents display behavioral patterns that are very different from being random, even in the absence of reward. Our aim is to understand foraging behavior by implementing an artificial rat that behaves as real ones do. We depart from the hypothesis that rodents, when performing free exploration, may be minimizing the unpredictability of the environment in terms of internally mapping its structure and discovering all the actions that it can afford. This drive for exploration is counterbalanced by a drive for security. Building from a self-regulation model based on the Distributed Adaptive Control architecture (DAC), we implement a biomimetic control that uses this predictability principle to generate behavior. We validate the controller by solving a benchmark task in which the agent learns to displace a movable obstacle to discover unexplored areas of an arena.

## Introduction

In this paper we take a behavioral approach to the deeper understanding of what it means to survive, explore and forage in complex environments, specially focusing on rodents. We provide a biomimetic control based on self-regulation that uses predictability as main principle to generate behavior, map the environment and discover the actions that the environment can afford. In a benchmark task the agent learns to displace a movable obstacle to discover unexplored areas. In free exploration, rodents exhibit a structured behavior following a specific pattern. If the environment is unknown, rats start the exploration following the walls, establishing a preferred corner and traversing the arena occasionally (Dvorkin et al., 2008). An example of rat trajectory in a squared arena can be seen in figure 1.

The question arises if this structured but still complex behavior can be explained by a minimal set of basic principles. Predictability has been exploited as a possible candidate for driving behavior and learning mechanisms of artifacts (Duff and Verschure, 2010; Weiller et al., 2010). As a more general principle, predictability has also been considered in the

free energy principle in the form of the minimization of surprise (Friston et al., 2006). In the free energy principle surprise is defined in statistical terms and can be minimized both by the behavior of the artifact as well as by adapting the internal model of the artifact.

All these approaches have a limited sense of the real consequences of what it means to act in a world. For instance, scenarios where the agent's actions can change the state of the external world, i.e. the state of objects in the environment, are not considered. Moreover, the described optimization algorithm (Weiller et al., 2010) and free-energy principle (Friston et al., 2006) are conceptually far from being biomimetic implementations and don't deal with fundamental physiological mechanisms such as self-regulation.

Here we explore predictability together with security as an internal drive for free exploration. Predictability in our case will depend on the agent ability to map the environment and discover the affordances available in it. Affordances, as introduced by Gibson (1986), are considered to represent all the action repertoires available to an agent in an environment. The more the agent learns about the possible actions in an environment and their consequences, the better it can predict the next sensory state he will be in. The environment will not be surprising any more when the agent knows its structure and knows what it can do in it.

The robotics community has also shown an interest for the predictability driven learning (also called curiosity) through object interactions (Oudeyer et al., 2007; Ugur et al., 2007). Our main contribution is to provide a biomimetic solution by introducing in a self-regulatory loop, the necessity of making the environment a predictable place. We build from previous work on the Distributed Adaptive Control (DAC) architecture (Verschure et al., 2003; Duff and Verschure, 2010). DAC provides a continuous sensorimotor loop combined with memory, organized in a layered structure. We have investigated affordances and the acquisition of sensorimotor contingencies in the context of DAC (Duff and Verschure, 2010). In (Sánchez-Fibla et al., 2010b) we have equipped the lowest layer of DAC, the reactive layer, with a self-regulatory process based on the physiological notion of

allostasis. Self-regulation was decomposed into a minimal number of homeostatic loops and allostasis was the meta-process that controlled stability of the system at a higher level, changing the desired values (the objectives to reach) of each subsystem. We validated the model by comparing the generated behavior with the one displayed by rodents in different environments. In the case of free exploration of a squared arena, only two subsystems were considered: security and arousal. Security was the subsystem controlling the distance of the agent to a familiar place, like the home-base, and arousal controlled the exposure to the open space. Each subsystem in the allostatic control was defined by a gradient (in accordance to motor schema based behaviors, see introductory book of behavior-based robotics by Arkin, R. C. (1998)), an actual value and a desired value. The security and arousal gradients were assumed and predefined in (Sanchez-Fibla et al., 2010b), here we adapt the complete sensorimotor loop to be able to learn those gradients as the agent explores the environment. To do so we directly link arousal to unpredictability following the principle that an unpredictable space will induce a higher state of arousal. The drive to explore/discover the environment is thus considered equivalent to the urge for higher arousal. Therefore behavior is driven by security and predictability of the environment being regulated inside a sensorimotor loop.

Building from (Sanchez-Fibla et al., 2010b), we state that exploration in this simple scenario is also driven by the agent's ability to predict the environment in two aspects: its structure and its affordances. To further develop this hypothesis we enrich a squared arena environment with the presence of an object, in our case a cube. To understand behavior it is important to include elements that allow an agent to exploit affordances. For this purpose, we validate

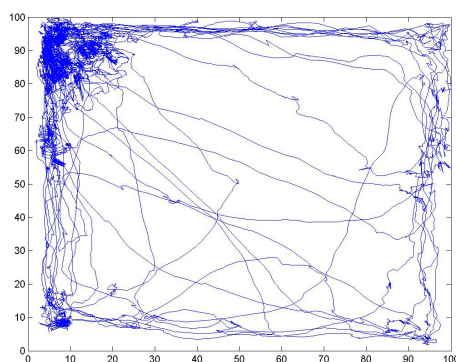


Figure 1: Trajectory plot of a rat. The plot displays the trajectory of a rat when performing free exploration in a squared arena. Axis units are in centimetres. According to the model that we presented in (Sanchez-Fibla et al., 2010b), this would correspond to a low aroused rat, in the sense that the traversals of the middle of the arena are scarce.

our controller with a benchmark task where an object in a squared arena is obstructing an alley that accesses a hidden room. The paper structure is as follows: first we describe in the methods section the allostatic control and how we modify it for the task that we want to solve. In the results section we compare the implemented controller based on minimizing unpredictability to a random controller. In the discussion we point out the links of the model to the behavioral studies of rodent

## Methods

In this section we describe the self-regulation mechanism driving the agent. We present a process for mapping the environment and discovering the available affordances as they are a prerequisite for minimizing the unpredictability of the environment. We then explain in more detail the arousal subsystem as it contains the main changes from previous studies (Sanchez-Fibla et al., 2010b) and (Sanchez-Fibla et al., 2010a).

### Allostatic control revised

We have proposed a biomimetic architecture of perception, cognition and behavior, called Distributed Adaptive Control (DAC) architecture, which aims at explaining how the interaction of different structures along the neuraxis can give rise to adaptive behavior (Verschure et al., 2003). In the latter, drive based behavioral control is modelled subserving perception and cognition from the perspective of the interaction of appetitive and aversively motivated behaviors. In this case the reactive regulation between these two orthogonal behavior tendencies was achieved through predefined rules for conflict resolution; i.e. aversion and avoidance supersedes consumption and approach. Hence, in this system the relationship between these two drive systems was fixed and could not be dynamically regulated.

In (Sanchez-Fibla et al., 2010b) we included in DAC the ability to regulate drive based behaviors with the objective to identify a solution that scales with respect to the number of behavioral subsystems, that provides a common currency for the regulation of behavior in order to unify multiple levels of a real-world cognitive architecture and that is biologically valid. Animals are driven by internal variables such as hunger, temperature, security, etc. which have to be maintained within certain limits in order to be stable and predictive over changing environments. We follow this behavioral driven top-down approach instead of modelling the low level interactions of homeostatic processes (as for example the homeostatic regulations happening in the Endocrine system (Moioli et al., 2009; Xu and Wang, 2011), which we could consider to be a more bottom-up approach than the one followed here). For this, we decomposed self-regulation into a minimal set of homeostatic subsystems (such as arousal, security, energy, etc.) that can be plugged together orchestrated by what we named allostatic control in (Sanchez-Fibla

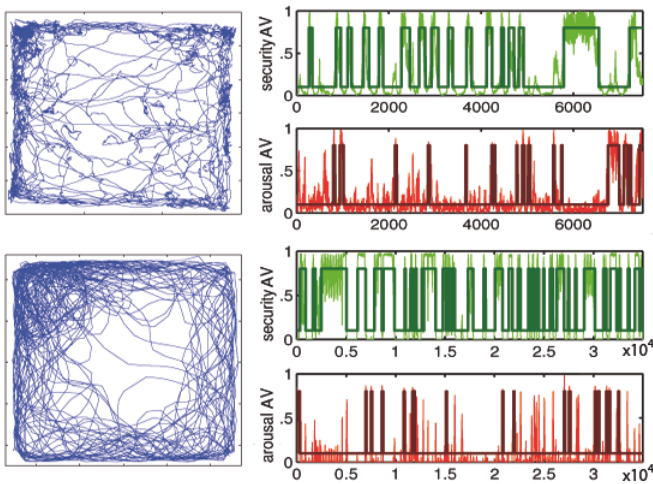


Figure 2: Rat-robot behavior comparison. On the top row we plot the trajectory of one rat along with its security  $AV$  time series (in green) and the arousal  $AV$  time series of the same session (below, in red). The bottom row corresponds to the generated behavior of the allostatic control system described in (Sanchez-Fibla et al., 2010b). First we plot the trajectory of the robot next to its security and arousal time series.

et al., 2010b). Each homeostatic loop consists of a gradient, an actual value ( $AV$ ) of the agent in that gradient, a desired value ( $DV$ ) and a regulator able to perform the appropriate actions to bring the actual value closer to the desired one. See for example the security subsystem in figure 3 where all these elements are shown. Objectives of different homeostatic loops may be conflictive, is in that stage that allostasis enters into play. We define allostasis as the regulation through changes of the desired values so that stability of individual homeostatic loops can be achieved or compromised and changing through time. In (Sanchez-Fibla et al., 2010b) we used a probabilistic changing policy of desired values also dependant of its level of content (the difference between  $DV$  and  $AV$ ).

The assumption in (Sanchez-Fibla et al., 2010b) was that behavior of a rat in a squared arena is driven by the constant equilibrium of its need for security (the distance to the home base or preferred corner) and the need for exploration conveyed by its need for arousal (exposure to the open space). An example of a trajectory generated by the model can be seen in figure 2, bottom row. The security gradient was maximum at the top left corner and the arousal was a fixed gradient having its maximum in the middle of the arena. The time series of the  $AV$  values of both subsystems are shown in the right and corroborate that the allostatic control system interleaves stays in the preferred corner with occasional traversals of the arena.

In (Sanchez-Fibla et al., 2010b), both the security and the

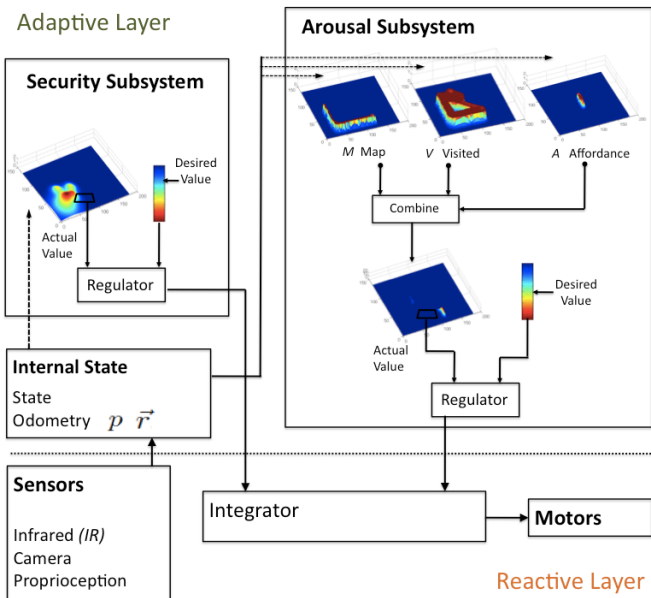


Figure 3: Allostatic control in DAC. The adaptive layer contains the main ingredients for the self-regulatory loop managed by the allostatic control. See text for further explanation.

arousal gradient are predefined. Here we learn the arousal gradient following the assumption that arousal is directly linked to unpredictability. Thus a high desired value for arousal corresponds to a high level of exploration. However when the rat explores the environment it is also increasing the predictability of the environment: meaning that its drives are influenced by its necessity of knowing the structure of the arena and all the possibilities that the environment offers in terms of affordable actions. Thus, we consider that the need of increasing arousal is directly related to the notion of curiosity.

As in the squared arena setup of (Sanchez-Fibla et al., 2010b), we continue to decompose self-regulation into two minimal subsystems, security and arousal, but now arousal is redefined as a conjunction of the need for mapping the structure of the arena and the need for discovering its affordances. Security is represented by a gradient  $S$  that is maximum in a preferred place of the agent (Sanchez-Fibla et al., 2010b). Arousal is calculated using three gradients that we now list;  $M$  the map gradient of the environment, the one which the agent uses to accumulate evidence of the presence of walls and obstacles.  $A$  the affordance gradient where the agent accumulates evidence where the environment affords an action,  $V$  the visited gradient that the agent fills during exploration. From these three gradients  $M$ ,  $A$  and  $V$  we build an arousal map that plays the role of a saliency map of places that remain worth visiting. The arousal gradient is the one used by the arousal subsystem. An schema of the DAC

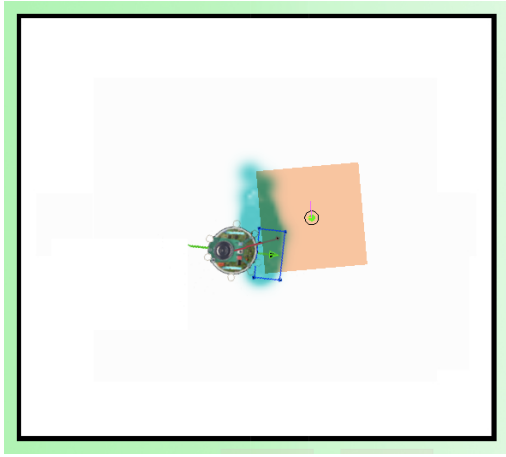


Figure 4: Simulation environment. A squared object originally placed in the middle of the arena, its being pushed by the epuck robot. Its "pushability" has been detected and the affordance gradient  $A$  has been updated accordingly.

architecture reduced to the allostatic control loop is shown in figure 3.

Original DAC is organized along three levels of control of increasing complexity: reactive, adaptive and contextual. We don't need contextual control for the tasks that we solve here. The reactive layer provides a pre-wired repertoire of reflexes. The adaptive layer acquires representations of sensory events and associated responses supporting the acquisition of simple tasks. In the adaptive layer we acquire the different internal structures like the gradients  $M$ ,  $A$  and  $V$ .

Security and arousal subsystems propose their motor actions, through a regulator process which computes a motor plan that could bring closer the actual value and the desired one. In (Sanchez-Fibla et al., 2010b), this computation was done having access only locally to the gradient around the position of the robot. This does not need to be so. We assume now that the gradients are sensory motor representations acquired by the adaptive layer and thus the regulator can have global access to its values. The outputs of both subsystems are then linearly summed (represented by the "Integrator" box of figure 3) and sent to the motors.

A difference from (Sanchez-Fibla et al., 2010b) is that the gradients that we use are computed during the exploration and not assumed and given to the model. In the following sections we explain how we compute them.

### Mapping the environment

We describe here how the agent builds the map gradient  $M$  while it explores the environment. When implementing controllers in a mobile robot, usually because of their limited sensor models, we face the difficulties of estimating the world state with their local sensing capabilities. Here we use the e-puck robot (Mondada et al., 2009), its infrared ( $IR$ )

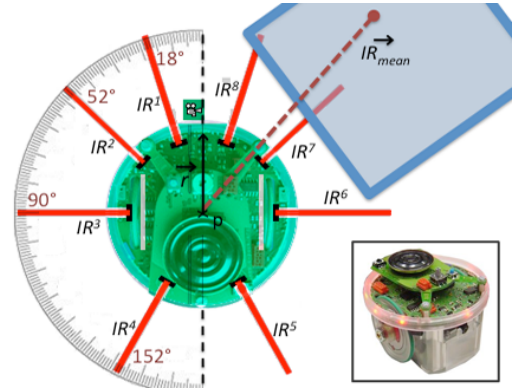


Figure 5: Top view of the e-puck sensor model. We indicate the position of the top/down camera and the angle in degrees of the 8 infrared sensors ( $IR^1 \dots IR^8$ ).  $IR^5 \dots IR^8$  have negative angles with respect to the front direction of the robot. For instance,  $IR^8_\alpha = -18^\circ$ . The direction and position of the robot are denoted by  $\vec{r}$  and  $p$ , respectively. We also plot a possible  $\vec{IR}_{mean}$  computed from the IR reading when approaching a squared object. See text for further explanation. In the bottom right part of the picture we show a general view of the e-puck robot.

sensors, indicated in figure 5 and motors. We assume the robot has an odometry model that can estimate its position and direction in the environment:  $p$  and  $\vec{r}$ . If we are thinking of a biomimetic solution, we could implement a grid cell system that would provide odometry plugged together with a place cell system that could synchronize to precise locations with the help of external cues, an approach that has been proven effective in (Milford and Wyeth, 2008). Also In (Wyss et al., 2006), a neuronal network is trained for acquiring place cell activity from one single camera input stream.

Using this information, we describe now how we compute the gradient that will capture the structure of the environment (see algorithm 1). Whenever the front  $IR$  sensors detect the presence of a wall or an object, the evidence of the presence of this object can be added to corresponding gradient  $M$ . In figure 5 we show the e-puck sensor model. Red lines represent the range of the  $IR$  sensors. The  $IR$  sensors have a decay and we have assumed is quadratic. This decay has to be compensated to correctly estimate the position of the border of an object. This compensation is computed in line 2 of the algorithm and then multiplied by an arbitrary grid unit constant (in this case 30). We denote  $IR_{val}^i$  the value of the  $IR$  sensor  $i$  normalized to take values from 0 to 1, being the latter the closest that an object can be sensed.  $IR_\alpha^i$  is the angle of the sensor with respect to the direction of the robot. The operator  $\angle$  used as super-index denotes the vector rotation operator. In line 3 we compute the point of contact by summing the radius of the robot and the computed distance in the direction of the  $IR$  sensor with the

agent position  $p$ . We then add a gaussian in the  $M$  gradient map at the computed point  $p_{map}$  with sigma  $\sigma$ .

---

**Algorithm 1: Mapping()**


---

```

for  $i = 1$  to  $4$  do
1  if  $IR_{val}^i > 0.65$  then
2     $dist \leftarrow 30(\frac{1-IR_{val}^i}{0.25})^2 \tilde{d}^{\angle IR_{\alpha}^i}$ 
3     $p_{map} \leftarrow p + (radius + dist)\|\tilde{d}\|$ 
4     $M \leftarrow M + g_{\sigma, p_{map}}$ 

```

---

The visited gradient  $V$  can be trivially filled by adding evidence in the agent position  $p$  at each time step. Due to objects placed in the environment that can act as obstacles when approached from one direction, and can be pushed from another, the agent can add in  $M$  evidence of presence of a wall when it is in fact not the case. For this reason, we add a decaying mechanism that uses the information in the visited gradient  $V$  to bring down to 0, areas in the  $M$  gradient where there was evidence of a wall but can in fact be visited and traversed.

### Affordances in the environment

We discuss in this section how an agent can discover the affordances of an environment. From the action repertoires that an object can potentially serve (its affordances), we restrict ourselves to the "pushability" of an object, that is, knowing when an object can be moved or it is a fixed part of the environment. In SLAM the agent learns about the world through passive exploration of the environment, that is, no real action of the agent can change the state of the world, except its position. In some cases dynamic obstacles are considered but their appearance or disappearance is not dependant on the agent's actions but governed by external interventions, as in (Kawewong et al., 2010). To extend SLAM to include the usage of affordances one could simply use another sensor modality to detect the sliding direction. One could think of using optical flow or tactile sensing. Sliding detection is a prerequisite for an agent to be able to distinguish from two situations: pushing unmovable objects and pushable objects. Detecting an object in front of the agent while having full forward speed would mean that the object is being pushed if there is a consistency of the signal sent to the motors and the detected optical flow main direction. In the other case where the agent detects an object in front, the motors are running forward but no consistent movement is detected by the optical flow would mean that we are sliding in front of a static obstacle/wall/object. In the custom e-puck simulator, we give global access to the agent to its real direction of movement. This vector could be computed, with the described mechanism using a camera pointing to a chessboard-tiled floor. The agent then compares this direction vector to the signal it is sending to the motors and can know when it is pushing a movable object or just being

blocked by an obstacle or a wall. This detection mechanism is used by the agent to fill gradient  $A$ . Whenever it detects that it is pushing an object, instead of accumulating evidence in the gradient map  $M$  (as in previous section), it does so in the affordance map  $A$ .

This accumulation is sufficient for the task that we want to solve here, which is guiding exploration. It would be not enough if we wanted to map the position of movable objects and know its position and orientation in the environment. For this purpose we have investigated in another paper in preparation, object-centred representations.

The agent can estimate the center of the object and its contour. There is a straightforward way of estimating the center and that is using a weighted mean of the vectors at each  $IR$  sensor direction. We call this vector  $\vec{IR}_{mean}$  and we show an example of it in figure 5 and also 4. We compute it with the following vector sum:

$$\vec{IR}_{mean} = \sum_{i=1}^8 IR_{val}^i \vec{r}^{\angle IR_{\alpha}^i}$$

Then the center of the object can be estimated by summing  $\vec{IR}_{mean}$  to the current agent position:  $p_{obj} \leftarrow p + \vec{IR}_{mean}$ .

The agent can also estimate the contour of an object by turning around it and mapping its border using the  $IR$  sensors and the described mechanisms.

### The arousal subsystem

The arousal subsystem combines the map  $M$ , the affordance  $A$  and the visited  $V$  gradients to generate another gradient that can be interpreted as a saliency map of places worth visiting, places where predictability of the environment can be increased. As  $M$ ,  $V$  and  $A$  are built while exploring the environment, the resulting gradient can only consider the visited gradient border. We cannot assume that the agent knows a part of the environment that has not been explored yet. For capturing the visited border we choose random points having  $V(i, j) > 0.01$ . If a random point satisfies this last condition we add a Gaussian weighted by a saliency value into the new gradient. The saliency value inversely depends on the sum  $\sum_{i,j} V(i, j)$  around a predefined radius. This will promote the fact that places that have not been visited yet have high saliency. We also weight the saliency by the sum of the map gradient  $\sum_{i,j} M(i, j)$  and  $\sum_{i,j} A(i, j)$ . This last condition considers parts next to the walls or objects that have not been mapped/pushed yet. We then give preference to points that are close to the agent current position. This combination of gradients to compute the resulting saliency is represented by the box "Combine" in figure 3.

## Results

We validate the arousal subsystem based on the predictability principle using a benchmark task designed for mobile



robotics (implemented in a custom simulation environment). The agent has to map and explore the totality of an environment in which a hidden alley has been obstructed by a pushable object. A snapshot of the used environment is shown in figure 6. It consists of a squared arena joined with a smaller rectangular chamber through a small alley. In the initial state, the squared object is placed obstructing the alley connecting to the hidden part of the arena, as shown in the figure. Superimposed in the arena we show the trajectory of the agent in one trial (dashed red line). The movement that the square displayed during the trial is plotted with a series of blue square contours.

To be able to benchmark different controllers solving the mapping task we introduce two measures:  $r_{mapped}$  and  $r_{visited}$ . These allow us to quantify how much the environment has been mapped and how much it has been explored respectively. The  $r_{mapped}$  and  $r_{visited}$  measures take values between 0 and 1 whether the whole environment has been mapped/visited, then  $r = 1$  or the agent has just been inserted in the environment  $r = 0$ .  $r_{mapped}$  is computed by counting how many points in the arena shape have activity in the map gradient  $M$  and then dividing by the total number of points. Similarly,  $r_{visited}$  is computed by counting the number of points in the environment that have been covered by  $V$  and dividing by the total number of points.

We define a random controller that allows us to benchmark the exploration and mapping capabilities of the described controller. The random controller builds an arousal saliency map (described in previous section) by choosing a random point in the border of the visited gradient.

In figure 7 we compare the  $r_{mapped}$  and  $r_{visited}$  measures for the random controller and the described arousal subsystem. One trial is sufficient to show that the environment is mapped quicker with the improved arousal subsystem. While in the random controller case,  $r_{mapped}$  and  $r_{visited}$  increase in a rather monotone constant way, in the improved controller case we observe a quicker increase in the begging until reaching a flat part between iterations 4000 and 7000 (corresponding to the explored square arena part). When the hidden alley is discovered the measures increase again. In the random controller case, the hidden chamber of the arena is not discovered during the shown 12000 iterations.

## Discussion

Little is known about rodent's ability to exploit affordances. Nevertheless it has been shown their ability to internally represent objects, proven by the fact that rats lose interest of objects which they have been able to interact and gain interest of objects which they did not encountered before. This is what it is called the novelty preference in (Ennaceur, 2010).

Experiments show that object properties are accessible to rodents along various dimensions such as shapes, textures, odour, color and brightness. Although this vast range of recognition characteristics, rats show a preference for ob-

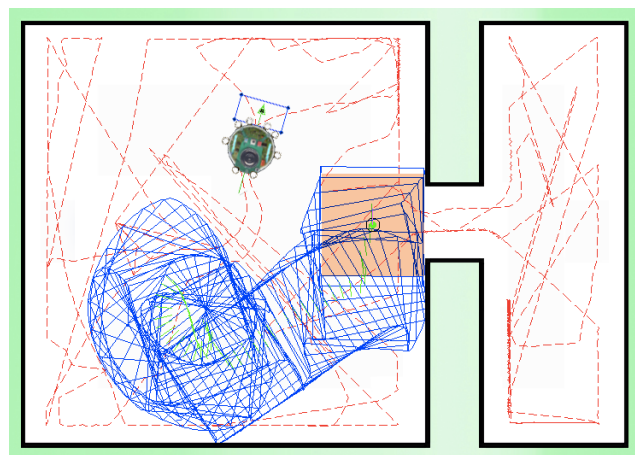


Figure 6: One trial of the robot task. The dashed red line corresponds to the trajectory of the robot. The blue squares correspond to the movement displayed by the squared object when pushed.

jects that have affordances for common rat activities, for example, in (Chemero and Heyser, 2005) it is found that rats prefer objects they could climb onto to those they could not. Similarly, rats show interest for the manipulation (e.g. grasping, pushing) of objects that could interfere with accessing new unexplored areas of the arena, alleys, corridors, and thus could be rewarding in a later state. These facts have informed the realization of the benchmark task that we solve in this paper and also have opened future directions for generating new benchmarks and experimental tasks using rodents.

It's worth noting the absence of reward function in our paradigm. This fact set us apart from classical Machine Learning approaches. Self-regulation is the responsible of building the reward function indirectly, combining all the objectives of the different subsystems.

Several neuronal based computational models exist that address the issue of reproducing rodent behavior focusing on navigation and relating it to the hippocampus, see for example (Sheynikhovich et al., 2009). We addressed this issue from this perspective in (Sanchez-Fibla et al., 2010a). This was not our aim in this research. Same applies to the mentioned research on homeostasis as a low level regulatory process present in the Endocrine System (Moioli et al., 2009; Xu and Wang, 2011). Here we wanted to focus on the unpredictability minimization principle that drives exploration introducing affordances as a new dimension to be considered.

## Conclusions

We have presented a biomimetic controller based on the self-regulation of two internal variables: security, defined as the distance to a familiar established place, and arousal, in terms

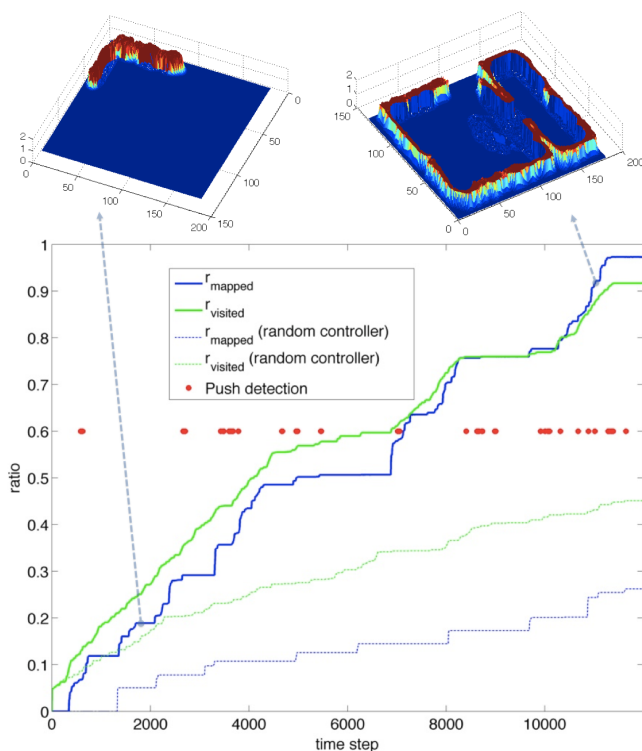


Figure 7: Simulation results. The x axis corresponds to the simulation time steps. Solid lines indicate the environment mapped measure (in blue) and the visited measure (green) for the modified allostatic control. Dashed lines correspond to the same measures for a random behaving controller. Red dots indicate the time steps when the agent detect that it was pushing an object (during the allostatic control session). Two snapshots of the gradient map being built are shown on top.

of the predictability of the environment. This implementation is based on the hypothesis that rodents, when exploring an environment, may be interleaving its need for security and the minimization of the unpredictability of the environment, in terms of internally mapping its structure and exploiting all the actions that it can afford. We use a simulated epuck robot to validate our controller and expose an agent to an environment with a squared object. In a final benchmark task, the object is obstructing an alley having access to a hidden part of the environment. We show that using the described controller the agent is able to map more rapidly the environment and the actions it can afford than a simple random controller.

A future step would be to compare the model with experimental data of rodents performing free exploration in environments that afford actions: push objects, push doors, etc.

## Acknowledgements

This work was supported by Synthetic Forager FP7-ICT-217148 and eSMC FP7-ICT-270212. We would like to thank partners in the eSMC consortium for helpful discussions. We thank Matti Mintz and Erez Wasserman from Tel Aviv University for providing us with the two rat sessions displayed in figures 1 and 2.

## References

- Arkin, R. C. (1998). *Behavior-Based Robotics*. MIT Press.
- Chemero, A. and Heyser, C. (2005). Object exploration and a problem with reductionism. *Synthese*, 147(3):403–423.
- Duff, A. and Verschure, P. (2010). Unifying perceptual and behavioral learning with a correlative subspace learning rule. *Neurocomputing*, 73(10-12):1818–1830.
- Dvorkin, A., Benjamini, Y., and Golani, I. (2008). Mouse cognition-related behavior in the open-field: emergence of places of attraction. *PLoS Comput Biol*, 4(2):e1000027.
- Ennaceur, A. (2010). One-trial object recognition in rats and mice: Methodological and theoretical issues. *Behavioural Brain Research*.
- Friston, K., Kilner, J., and Harrison, L. (2006). A free energy principle for the brain. *Journal of Physiology-Paris*, 100(1-3):70–87.
- Gibson, J. (1986). *The ecological approach to visual perception*. Lawrence Erlbaum.
- Kawewong, A., Tongprasit, N., Tangruamsub, S., and Hasegawa, O. (2010). Online and Incremental Appearance-based SLAM in Highly Dynamic Environments. *The International Journal of Robotics Research*.
- Milford, M. and Wyeth, G. (2008). Mapping a suburb with a single camera using a biologically inspired SLAM system. *Robotics, IEEE Transactions on*, 24(5):1038–1053.
- Moioli, R., Vargas, P., and Husbands, P. (2009). A multiple hormone approach to the homeostatic control of conflicting behaviours in an autonomous mobile robot. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 47–54. IEEE.
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptoch, A., Magnenat, S., Zufferey, J., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions*, volume 1, pages 59–65.
- Oudeyer, P., Kaplan, F., and Hafner, V. (2007). Intrinsic motivation systems for autonomous mental development. *Evolutionary Computation, IEEE Transactions on*, 11(2):265–286.
- Sanchez-Fibla, M., Bernardet, U., and Verschure, P. (2010a). Allostatic control for robot behaviour regulation: An extension to path planning. In *Int. Conf. on Robots and Systems, IROS*, pages 1935–1942. IEEE/RSJ.



- Sanchez-Fibla, M., Bernardet, U., Wasserman, E., Pelc, T., Mintz, M., Jackson, J. C., Lansink, C., Pennartz, C., and Verschure, P. (2010b). Allostatic Control for Robot Behavior Regulation: a Comparative Rodent-Robot Study. *Advances In Complex Systems*, 13(3):377–403.
- Sheynikhovich, D., Chavarriaga, R., Strosslin, T., Arleo, A., and Gerstner, W. (—2009—). Is there a geometric module for spatial orientation? insights from a rodent navigation model. *Psychological Review*, 116(3):540–566.
- Ugur, E., Dogar, M., Cakmak, M., and Sahin, E. (2007). Curiosity-driven learning of traversability affordance on a mobile robot. In *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*, pages 13–18. IEEE.
- Verschure, P. F., Voegtlin, T., and Douglas, R. J. (2003). Environmentally mediated synergy between perception and behaviour in mobile robots. *Nature*, 425(6958):620–4.
- Weiller, D., Martin, R., Dähne, S., Engel, A., König, P., and Krichmar, J. (2010). Involving Motor Capabilities in the Formation of Sensory Space Representations. *PloS one*, 5(4):607–609.
- Wyss, R., König, P., and Verschure, P. F. M. J. (2006). A Model of the Ventral Visual System Based on Temporal Stability and Local Memory. *PLoS Bio*, 4(5):e120.
- Xu, Q. and Wang, L. (2011). Recent advances in the artificial endocrine system. *Journal of Zhejiang University-Science C*, 12(3):171–183.

# Swarms for Robot Vision: The Case of Adaptive Visual Trail Detection and Tracking

Pedro Santana<sup>1,2</sup>, Ricardo Mendonça<sup>1</sup>, Luís Correia<sup>2</sup> and José Barata<sup>1</sup>

<sup>1</sup>UNINOVA, Universidade Nova de Lisboa, Portugal

<sup>2</sup>LabMAg, Universidade de Lisboa, Portugal  
pfs@uninova.pt

## Abstract

Previous work has shown that a pheromone-based visual saliency map can be computed by a swarm of simple agents inhabiting the robot's input image. It was also shown that, with a proper set of behaviours controlling the agents, the saliency map can be used to localise trails present in the robot's visual field. Under the assumption that the robot starts its autonomous operation already on the trail, this paper extends that work by enabling the agents to learn online an appearance model of the trail. The learned model is then used to increase the level of pheromone deployed in the regions of the input image that are more probable of belonging to the trail. This is motivated by the well-known importance that a priori object knowledge has to improve visual search. The outcome of this extension is a self-organising behaviour capable of detecting trails in 98% of the evaluated situations, outperforming the original work. The agents being simple their computation is fast, resulting in a 12 Hz performance. Thus, by introducing a parsimonious learning mechanism, this paper contributes to increase robustness of swarm-based robot vision systems.

## 1. Introduction

An important sensory modality for autonomous robots is vision. However, the richness of vision comes with the price of complex processing. The complexity inherent to vision calls for fine and contextualised focus of computational resources on the most relevant stimuli obtained from the environment. This process is called visual attention, which has been extensively studied in humans (Oliva and Torralba, 2007; Wolfe et al., 1989). By focusing perception: (1) computation, and consequently, energy are more efficiently used; (2) the robot becomes less sensitive to noise and perceptual aliasing; and as a consequence of the previous two, (3) faster robot motion, lower cost, and reduced robot size are enabled.

Models of visual attention typically assume the existence of a sensory-driven bottom-up pre-attentive component (Treisman and Gelade, 1980; Itti et al., 1998), which is modulated by top-down context aware pathways (Tsotsos et al., 1995; Neider and Zelinsky, 2006). The use of top-down modulation is important when bottom-up saliency information is insufficient to focus attention in the presence

of distractors. These distractors are other objects or perceptual aliasing in the environment that happen to detach from the background at least as much as the object being sought. However, top-down information (e.g., expected colour and morphology of the object) is quite dependent on the environmental context. As a result, adapting this knowledge is key when facing unstructured environments.

Visual attention ultimately drives the motion of sense organs, e.g., eyes, towards the relevant stimulus source. This is called overt attention. A faster process is the one of mentally focusing particular aspects of the sensory stimuli. This is called covert attention and its modelling is the focus of this work. Studies on human subjects support the hypothesis that multiple covert attention processes co-exist in the brain (Doran et al., 2009).

In previous work (Santana and Correia, 2010, 2011), we have explored the idea of existing multiple covert attention processes to model visual attention on autonomous robots as the product of a self-organising process supported by a set of virtual agents inhabiting the sensorimotor space of the robot. In particular, we have devised a model where the action selection process is used as top-down context knowledge to guide visual obstacle detection. In that work, agents perform local covert visual attention loops, whereas the self-organising collective behaviour maintains global spatio-temporal coherence. In a related research line (Santana et al., 2010), we have shown that a swarm of agents is able to create saliency maps using implicit knowledge about the object being sought. The model was shown to be able to detect and track trails in natural environments. This top-down knowledge was defined in terms of the behaviours controlling the agents. Focusing on the shape of trails, rather than in their photometric appearance, is advantageous given trails variability. However, photometric appearance may be useful to compensate for situations where shape information is not reliable. Due to its variability under different contexts, photometric appearance must be considered under an adaptive framework, capable of being tuned to the specificities of the environment. The current paper addresses this problem by including an adaptive mechanism into the agents compos-

ing the swarms responsible for the localisation and tracking of the trail. Concretely, the output generated by the swarm in previous frames is used to supervise the learning process of a trail’s appearance model. In turn, this model is used to modulate the pheromone deployed by the agents, thus helping them concentrate their activity on the image regions whose appearance is more similar to the one of the trail being tracked.

## 2. System Overview

Typically, object-related a priori knowledge is used by top-down boosting of the set of features (e.g., colour) known beforehand to be more representative of the object being sought. Instead, the object’s overall layout, which is a more stable and predictable feature in the case of natural trails, whose local appearance often blends with the background, is used in this work. This type of a priori knowledge is specified indirectly in the proposed model as perception-action rules controlling the behaviour of simple agents inhabiting the robot’s visual input. These agents are called p-ants (from perceptual-ants) and represent local covert attention processes. Their self-organising collective behaviour results in a saliency map of the input image, and thus, in a global covert attention process.

Fig. 1 depicts the base model (Santana et al., 2010) of this work. In short, at each new frame **I**, two conspicuity maps,  $\mathbf{C}^C \in [0, 1]$  for colour and  $\mathbf{C}^I \in [0, 1]$  for intensity information, are computed (Santana et al., 2010). The intensity of a pixel in a given conspicuity map signals how much the pixel detaches from the background at several scales (i.e., resolutions), in the scope of a given visual feature. A set of  $n$  p-ants is then deployed on each map. These p-ants interact based on the ant-foraging metaphor for several iterations in order to build two pheromone maps,  $\mathbf{P}^C \in [0, 1]$  and  $\mathbf{P}^I \in [0, 1]$ . The behaviour of these p-ants is designed to exploit some a priori knowledge about typical trails approximate layout. The activation of the pheromone maps is therefore expected to match the trail’s location better than the activation of the conspicuity maps, which are only sensory driven. Additionally, by allowing p-ants on a given pheromone map to also affect the other pheromone map, cross-modality influences are implicitly, i.e., through stigmergy (Grassé, 1959), maintained in the system. This increases robustness by allowing p-ants to exploit multiple cues indirectly, in a simple and fast to compute way.

Rather than blending both conspicuity maps to generate the final saliency map  $\mathbf{S} \leftarrow \frac{1}{2}\mathbf{C}^I + \frac{1}{2}\mathbf{C}^C$ , as typically done (Itti et al., 1998), in this work  $\mathbf{S}$  is obtained by blending both pheromone fields,  $\mathbf{S} \leftarrow \frac{1}{2}\mathbf{P}^I + \frac{1}{2}\mathbf{P}^C$ . This way the saliency map is no longer a result of purely bottom-up sensory-driven process; instead, the bottom-up information is exploited under the context of some a priori knowledge about typical trails approximate layout. The result is a more robust and accurate focus of attention at the cost of a residual computa-

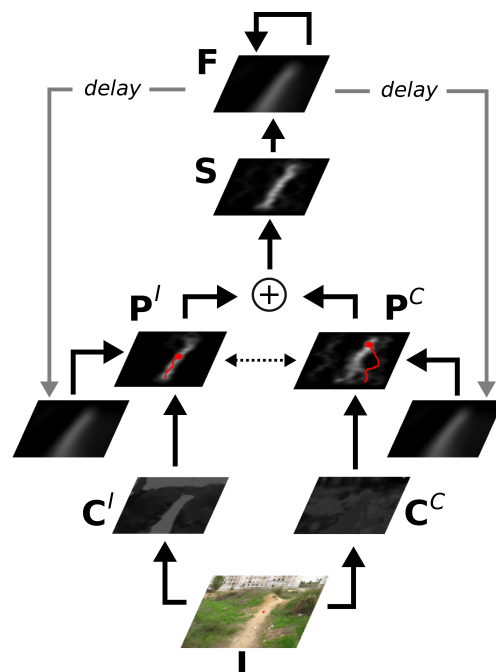


Figure 1: System’s operation overview (Santana et al., 2010). The red overlays in both pheromone fields,  $\mathbf{P}^C$  and  $\mathbf{P}^I$ , are two illustrative p-ant paths. Motion compensation aspects are not represented. Note that the brightest region in the neural field,  $\mathbf{F}$ , correctly corresponds to the trail location in the input image,  $\mathbf{I}$ .

tional overhead.

For across-frames integration of trail location evidence, the final saliency map  $\mathbf{S}$  feeds a dynamic neural field,  $\mathbf{F} \in [0, 1]$ , that is, a 2-D lattice of dynamical neurons with Mexican-hat shaped lateral coupling (Amari, 1977). This coupling implements inter-neuron local lateral excitation and long-range inhibition, which helps the neural field on the production of a single focus of attention (Rougier and Vitay, 2006). In order to decouple the dynamics of the neural field from the dynamics of the robot, the projective transformation estimated between frames is applied to the neural field. Finally, the output of the system is given by the current state of the neural field, in which the higher the activation of a given neuron the higher its chances of being associated to a trail’s pixel (refer to (Santana et al., 2010) for details on dynamical field processing).

In order to allow p-ants’ creation and activity to be affected by history, at the onset of each frame, both pheromone maps are initialised with a small ratio  $\lambda$  of the neural field after being motion compensated,  $\mathbf{P}^I \leftarrow \lambda \mathbf{F}$ ,  $\mathbf{P}^C \leftarrow \lambda \mathbf{F}$ . This induces stability and robustness to noise and temporarily mis-behaved conspicuity maps (i.e., unable to properly discern between trail and background in the presence of distractors), as well as it enables across-frames progressive im-

provement.

With the purpose of reducing the effects of strong distractors when tracking the trail, this paper includes into the swarm-based system an adaptive mechanism. The goal is to learn and update in each frame a simple appearance model of the trail, so that p-ants can strengthen the deployment of pheromone on regions whose appearance match the learned one. The result is a stronger stigmergic behaviour around the true location of the trail. Learning occurs by sampling the region of the visual input corresponding to the region of the neural field with highest activity. That is, the model is updated under the assumption that the trail location estimated in the previous frame is correct.

### 3. Pheromone Maps Computation

This section describes how the two pheromone maps,  $\mathbf{P}^I$  and  $\mathbf{P}^C$ , are built from the two conspicuity maps,  $\mathbf{C}^I$  and  $\mathbf{C}^C$ . For this purpose, a given p-ant,  $p_m$ , is created and associated to a given visual feature  $m \in \{I, C\}$ . The other visual feature is represented by  $m'$ . While being iterated for  $\eta$  times,  $p_m$  will move on  $\mathbf{C}^m$ , influenced by the pheromone present in  $\mathbf{P}^m$ . In the non-adaptive model, while moving, this p-ant deploys pheromone in each position visited in  $\mathbf{P}^m$  with a magnitude  $\epsilon_0$ , and a small portion of  $\epsilon_0$ ,  $v$ , in  $\mathbf{P}^{m'}$ .

After the iterations for this p-ant, a p-ant associated to the other visual feature,  $p_{m'}$ , is created and iterated following the same procedure. Afterwards, the two p-ants are removed from the system and the process is repeated  $n$  times, meaning that  $2n$  p-ants are created and iterated. As it will be shown, the deployed pheromone is a function of p-ants' sensations across their trajectories on their associated conspicuity maps. Hence, it is influenced by the activity occurring in distant regions of the map. This long-range spatial connectivity allows handling the potentially large size of trails in a robust and parsimonious way.

#### 3.1. P-Ant's Creation

The chances of creating a p-ant  $p_m$  on a given location  $\mathbf{o}_{p_m}$  of the conspicuity map  $\mathbf{C}^m$  depends on the level of conspicuity at that location and on the level of pheromone at the same location in the corresponding pheromone map,  $\mathbf{P}^m$ . Hence, p-ants are progressively and probabilistically deployed where there are more chances of being a trail, under the assumptions that: (1) trails tend to be conspicuous; (2) the trail has been successfully detected in the previous frame (represented by the feedback provided by the delayed neural field state); and (3) that the pheromone accumulated by p-ants deployed in the current frame builds-up mostly around the actual trail's location.

By assuming that trails often start from the bottom of the image, p-ants are deployed with a small randomly selected offset  $z \in [0, 0.1 \cdot h]$  of the bottom of the conspicuity map in question, i.e., at row  $r \in [h - z, h]$ , where  $h$  is the height

of the map<sup>1</sup>. This random small offset reduces sensitivity to any noise potentially present at the map's boundaries.

In order to determine the column where  $p_m$  is deployed, a unidimensional vector  $\mathbf{v}^m = (v_0^m, \dots, v_w^m)$  is first computed. The element  $v_k^m$  of  $\mathbf{v}^m$  refers to the average conspicuity level of the pixels in a small window centred on column  $k$  and with a randomly selected offset with respect to the bottom row of the map,  $r$ ,

$$v_k^m = \sum_{l,j} \frac{\mathbf{C}^m(l,j)}{\delta_w \cdot \delta_h} \quad (1)$$

where  $l \in [k - \delta_w/2, k + \delta_w/2]$ ,  $j \in [r - \delta_h, r]$ ,  $\mathbf{C}^m(l,j)$  returns the conspicuity level in position  $(l,j)$ , and  $\delta_w$  and  $\delta_h$  are the width and the height of the window, respectively. The same windowing process is applied to build a vector for the pheromone field in question,  $\mathbf{u}^m = (u_0^m, \dots, u_w^m)$ . Element  $u_k^m$  corresponds to the maximum pheromone level found in the window:

$$u_k^m = \max\{\mathbf{P}^m(l,j)\}_{l,j} \quad (2)$$

where  $\mathbf{P}^m(l,j) \in [0,1]$  returns the pheromone level in position  $(l,j)$ . The max operator is employed to benefit those regions where the paths of p-ants overlap more often and consequently where there is a higher consensus on the trail's skeleton position.

Using these two vectors in the following test, which is repeated until it succeeds, the chances of deploying a p-ant in a randomly selected column  $z_2 \cdot w$  is as high as the conspicuity and pheromone levels at the deployment region,

$$z_1 < \left( \rho \cdot u_{z_2 \cdot w}^m + (1 - \rho) \cdot v_{z_2 \cdot w}^m \right) \quad (3)$$

where  $z_1 \in [0,1]$  and  $z_2 \in [0,1]$  are numbers sampled from a uniform distribution each time the test is performed and  $\rho$  is a weight factor used to trade-off the influence of both pheromone and conspicuity information. By starting with a small value,  $\rho_0$ , and by linearly growing at each iteration by an amount  $\Delta\rho$ ,  $\rho$  operates as an adaptive process, compelling the system to move from a conspicuity-driven operation (exploration) to a pheromone-driven operation (refinement/exploitation).

#### 3.2. P-Ant's Execution

Before specifying p-ants behaviours, it is necessary to specify their sensory and action spaces. To reduce both sensitivity to noise and computational cost, the sensory input is defined by 5 coarse receptive fields disposed around the p-ant's current position,  $R_1 \dots R_5$  (see Fig. 2). For a given visual feature  $m$  and p-ant's position  $\mathbf{o}_{p_m}$ ,

<sup>1</sup> Rows are indexed in increasing order from the top to the bottom of the map.



iteration a p-ant  $p_m$  selects its action by maximising the following utility function, which incorporates behaviours' votes, pheromone-based interactions, and random fluctuations,

$$a_{p_m} = \arg \max_{a \in A} \left( \sum_{b \in B} \alpha_b f_b(p_m, a) + \mathbf{P}^m(R_a, \mathbf{o}_{p_m}) + \gamma q \right)$$

where:  $\alpha_b$  is a user defined weight accounting for the contribution of behaviour  $b \in B$ ; and  $\gamma$  is the weight accounting for stochastic behaviour, being  $q \in [0, 1]$  a number sampled from a uniform distribution each time the action is evaluated. To match the randomness magnitude with the scale of the image, which is typically smaller for pixels in upper regions of the image, the weight  $\gamma$  starts with an initial value  $\gamma_0$  and exponentially decays by a constant factor  $\gamma_\tau$  at each iteration.

In case an immediate loop is detected, namely, the p-ant moving recurrently from one pixel to another, then the action for the current iteration is randomly selected. Finally, the p-ant's position  $\mathbf{o}_{p_m}$  is updated according to the selected action<sup>2</sup>.

## 4. Adaptive Process

This section describes how (see Section 4.1) and when (see Section 4.2) the appearance model of the trail is learned and updated. To help p-ants disambiguate in situations where the conspicuity information is not sufficient by itself, the learned model is used to promote the deployment of pheromone on regions of the image whose appearance is more likely to belong to the one of the trail (see Section 4.3). To allow learning the model from scratch, some assumptions regarding the initial position of the trail with respect to the robot are made (see Section 4.4).

### 4.1. Appearance Model

The trail's appearance model of the current frame is a simple colour histogram,  $\mathbf{h}$ , of the pixels in the region of higher neural field's activity. To reduce sensitivity to illumination effects, the HSV colour space is used. To further reduce this sensitivity, the H(ue) component is described by 12 bins, the S(aturation) component by only 8 bins, and the V(alue) component is discarded altogether.

This frame-wise appearance model is used to update an across-frames appearance reference model,

$$\mathbf{h}_{\text{ref}} \leftarrow \Theta(\mathbf{F})\mathbf{h}_{\text{ref}} + 1 - \Theta(\mathbf{F}) \mathbf{h} \quad (5)$$

where  $\Theta(\mathbf{F}) = \kappa \cdot \max(\mathbf{F})$  makes the speed the reference model adapts to changes in the trail's appearance proportional to the neural field's maximum activity. This weighted

<sup>2</sup>For the sake of completeness, the pseudo-code of the models here described can be found at: <http://www.uninova.pt/~pfs/ecal2011trail.html>

approach allows the appearance model to be updated more strongly when the system is more sure of its output being a correct segmentation of the trail from the background. This assumption follows from the fact that the more stable the pheromone maps' activity across-frames the higher the neural field's maximum. Hence, the presence of distractors is less prone to affect the reference appearance model.

### 4.2. When to learn

To further reduce the chances of learning erroneous appearance models due to the presence of distractors, the appearance reference model,  $\mathbf{h}_{\text{ref}}$ , is only updated with Eq. 5 if the neural field in the current frame reports the trail as being roughly located ( $\pm 10\%$  of the map's width) at the centre of the image. This is a reasonable heuristic under the assumption that the robot is actively centring itself along the trail in order to follow it.

This learning gating process allows the reference model not to learn the appearance of transient distractors appearing in the sides of the trail. Furthermore, it allows the system to delay the learning phase when the robot does not start centred on the trail.

### 4.3. Adaptive pheromone deployment

In Santana et al. (2010), p-ants deploy a constant level of pheromone along their paths,  $\epsilon_0$  (see Section 3). In this work, instead, a given p-ant  $p_m$  deployed in map  $m$  deposits a non-fixed level of pheromone,

$$\epsilon = \epsilon_0 + \beta \cdot p(T|V_{p_m}) \quad (6)$$

where  $\beta$  is an empirically defined weighting factor and  $p(T|V_{p_m})$  is the probability of the p-ant's path,  $V_{p_m}$ , to belong to the trail (T).

The probability  $p(T|V_{p_m})$  is approximated by the average probability of pixels visited by the p-ant of belonging to the trail. These pixels are represented by the set  $V_{p_m}$ , and their individual probabilities are obtained directly from the normalised histogram  $\mathbf{h}_{\text{ref}}$ , according to a technique known as histogram back-projection (see Fig. 3 for typical results). As the experimental results will show, this simple approach suffices to help p-ants tracking the trail.

### 4.4. The First Frames

The advantages of using learning comes at the price of solving the bootstrapping problem. That is, in the absence of a learned model, the detector has a reduced chance of generating a good output to supervise the learning, which in turn hampers the learning of the model altogether. To solve this problem we start from the assumption that the detector is turned on when the robot is already roughly located on the trail. Therefore, we can assume that in the first frames the trail is centred on the robot's input image.

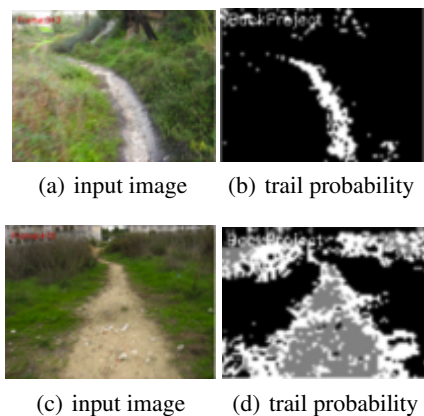


Figure 3: Pixel-wise trail probability (brightness level) for two typical images.

Bearing this in mind, in the first frames, instead of considering the maps' entire width,  $w$ , when selecting the deployment column of a newly created p-ant (see Eq. 1 and Eq. 2), the adaptive model assumes that the deployment region is constrained by a band centred on the map and with a frame-wise upper-bounded growing width. Concretely, in the first frame, the width of the band is 10% of  $w$ . Then, this width is increased by 0.5% at each new frame until the upper-bound  $w$  is reached. From then on, it remains static. At this moment the learned model is sufficiently mature to help the detector tracking the trail.

## 5. Experimental Results

This section quantifies the improvement the adaptive mechanism brings to the overall method and how well it suits the fast computation requirements imposed by physical robots. In order to measure the performance of the adaptive model, we relied on the same data-set used to evaluate the non-adaptive model (Santana et al., 2010). This data-set consists of 25 colour videos, encompassing a total of 12023 frames with  $640 \times 480$  resolution, which have been obtained with a hand-held camera<sup>3</sup>. This camera was carried at an approximate speed of  $1 \text{ ms}^{-1}$ . The trail detector was evaluated on an Intel T4300 2.1 GHz dual core, running Linux. OpenCV was used for low-level routines. To handle the probabilistic nature of the agents behaviour, a set of 5 runs was performed per video. In some of these videos the robot does not start on the trail, which is important to validate the ability of the detector to delay the learning phase.

Performance is measured as the percentage of frames in which the biggest blob of neural field activity above 0.85 (from a maximum of 1) is fully within the trail boundaries. The system parameters related to the adaptive mechanism,  $\kappa$ ,  $\beta$ , and  $\epsilon_0$ , have been empirically set to 0.001, 0.01, and

<sup>3</sup>The model's output overlaid on these videos is available at: <http://www.uninova.pt/~pfs/ecal2011trail.html>

0.008, respectively. The remainder of the free parameters have been set as in the original model (Santana et al., 2010).

With a success rate of  $92.98\% \pm 0.16\%$  over the 25 videos, the base model already attains an impressive result, operating  $\approx 4$  times better than a classical saliency model and in situations where previous detectors fail (details in (Santana et al., 2010)). However, a single failure in an embodied setup may result in dramatic consequences. Therefore, full success must be pursued. With the adaptive mechanism, the model reaches a success rate of  $97.94\% \pm 0.17\%$  over the 25 videos, and a 100% success rate in 12 of the 25 videos (see Table 1). Conversely, the non-adaptive model obtains a 100% success rate only in 6 of the 25 videos. Fig. 4 shows frames from some videos belonging to the 25 video data-set where the non-adaptive model fails to detect the trail, whereas the adaptive one succeeds. Although typically transient, these failures could drive the robot off trail. They usually occur when the assumption that trails are conspicuous structures fails due to the overall scene configuration. Sometimes it also happens that a sudden camera motion is not captured by the motion detection method, resulting in a mismatch between the neural field and the environment.

In terms of computation time, the non-adaptive model runs at 13 Hz whereas the adaptive one runs at 12 Hz. Note that only roughly 8% of the computation time refers to swarm-based activity - the remainder includes robot motion estimation, neural field update, and conspicuity maps computation. The conclusion is that the adaptive mechanism, which improves the method's accuracy, adds little computational overhead.

It is important to point out that the dependency of the overall process on an appearance model makes the learning process a critical one. This is reflected on the need for a learning bootstrapping process and for trail's appearance transitions to be smooth. That is, the improvement in performance is obtained at the cost of introducing assumptions, which are, nevertheless, acceptable under a trail tracking framework.

## 6. Discussion

Rather than static structures, like neurons, agents are better viewed as active information particles that flow and change in the system. Hence, using agents, the design focus is on the process and not so much on its supporting substrate. Additionally, agents being sensorimotor coordinated units can exploit the benefits of active vision (Bajcsy, 1988; Ballard, 1991) at the information processing level. These include the ability of agents to actively select and shape their sensory input so as to increase noise-to-signal ratio and increase their discriminatory power, to augment rotation and scale invariance, and also to exploit sensorimotor history with the purpose of inducing long-range influences and in the limit of improving their own behaviour (Scheier et al.,



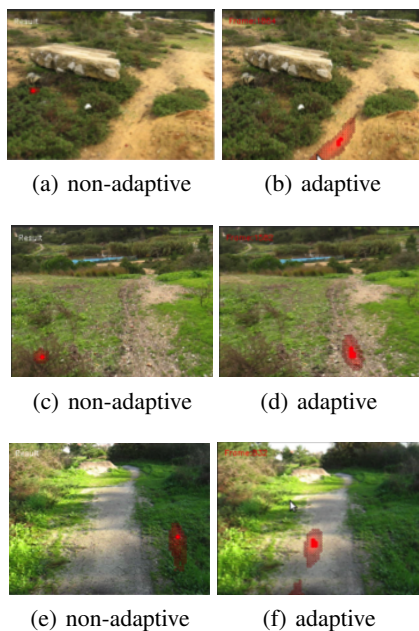


Figure 4: Examples of situations where the adaptive method outperforms the non-adaptive method. The red blobs represent the estimated trail location, which corresponds to the neural field activity above 85% of its maximum. In the adaptive case, besides localising the trail, the red blob is well aligned with its orientation. This means that the system is able to output both position and orientation of the trail.

Video ID	Nr. of frames	Non-adaptive model correct frames [%]	Adaptive model correct frames [%]
1	278	100.00 $\pm$ 0.00	100.00 $\pm$ 0.00
2	204	100.00 $\pm$ 0.00	100.00 $\pm$ 0.00
3	422	93.03 $\pm$ 0.21	99.15 $\pm$ 0.13
4	135	100.00 $\pm$ 0.00	100.00 $\pm$ 0.00
5	2854	93.90 $\pm$ 0.02	97.79 $\pm$ 0.03
6	186	97.53 $\pm$ 0.29	95.91 $\pm$ 0.48
7	121	100.00 $\pm$ 0.00	100.00 $\pm$ 0.00
8	124	88.06 $\pm$ 0.36	100.00 $\pm$ 0.00
9	309	98.38 $\pm$ 0.32	95.79 $\pm$ 0.32
10	147	92.11 $\pm$ 0.61	97.41 $\pm$ 1.12
11	386	100.00 $\pm$ 0.00	100.00 $\pm$ 0.00
12	158	88.48 $\pm$ 0.28	100.00 $\pm$ 0.00
13	134	87.31 $\pm$ 0.53	100.00 $\pm$ 0.00
14	676	99.14 $\pm$ 0.07	98.46 $\pm$ 0.17
15	683	91.22 $\pm$ 0.10	91.51 $\pm$ 0.10
16	770	82.96 $\pm$ 0.14	86.83 $\pm$ 0.30
17	403	93.90 $\pm$ 0.14	94.14 $\pm$ 0.83
18	335	86.21 $\pm$ 0.13	98.81 $\pm$ 0.30
19	230	76.43 $\pm$ 0.19	100.00 $\pm$ 0.00
20	439	82.92 $\pm$ 0.23	95.54 $\pm$ 0.38
21	490	93.31 $\pm$ 0.09	100.00 $\pm$ 0.00
22	230	100.00 $\pm$ 0.00	100.00 $\pm$ 0.00
23	600	90.10 $\pm$ 0.15	100.00 $\pm$ 0.00
24	802	95.06 $\pm$ 0.07	99.10 $\pm$ 0.10
25	907	94.42 $\pm$ 0.06	98.10 $\pm$ 0.09
<b>Total</b>	<b>12023</b>	<b>92.98 <math>\pm</math> 0.16</b>	<b>97.94 <math>\pm</math> 0.17</b>

Table 1: Comparative results summary.

1998; Nolfi and Marocco, 2002; Beer, 2003; Floreano et al., 2004; Sporns and Lungarella, 2006; Mirolli et al., 2010). Furthermore, the use of multiple agents in the task of modelling cognitive behaviour exploits biological knowledge obtained from similar processes that can be found in Nature.

In our line of research, we have used a model inspired by *swarm cognition* of social insects, whose considerable similarities with brain cognitive function are becoming widely recognised (Passino et al., 2008; Couzin, 2009; Marshall and Franks, 2009; Trianni and Tuci, 2011; Santana and Correia, 2010; Turner, 2011; Trianni et al., 2011). In this work, the ant foraging metaphor previously used was extended with learning capabilities, resulting in a system that can better adapt to different environmental contexts.

The use of learned appearance models to swarm-based object tracking has already been explored in the context of PSO-based models (Zhang et al., 2008). However, our work is the first applying learning to the problem of swarm-based trail detection and tracking. This is an important difference as the appearance of trails change more drastically than the one of typical objects. Furthermore, our model uses the appearance model to modulate pheromone deployment, a concept inexistent in PSO models.

## 7. Conclusions

This article proposes a model to incorporate an adaptive mechanism into a swarm-based trail detector previously published. The goal of this mechanism is to allow the detector to learn and exploit appearance models of the trail being followed. Experimental results confirmed the ability of the adaptive model to outperform the non-adaptive one, under the assumption that the robot starts its operation already on the trail.

The learned trail’s appearance model is used to modulate the swarm operation, rather than, to directly classify the input image as in a convolution-like typical computer vision operation. First, this approach allows the system to exploit synergistically both appearance and shape information, which is pivotal to handle sudden trail’s appearance changes. Second, this multi-modal approach allows the use of simple appearance models, i.e., histograms. Third, the appearance model and the behaviours controlling the agents being simple enable a fast to compute system.

With a bottom-up self-organising approach, the model is capable of handling highly unstructured trails without exhibiting a high computational load. In fact, we have shown in previous work (Santana et al., 2010) that the non-adaptive model performs in situations where previous detectors employing classical computer vision techniques would fail. In this work, we have improved the previous model by introducing elementary learning of the photometric appearance of the trail. All this leads us to conclude that swarm-based models are an interesting alternative to classical computer vision techniques. This means that besides contributing with

a useful model to improve off-road robot navigation, this work intends to encourage the artificial life community to employ their bulk of knowledge at the service of the high impact problem of synthesising robust and fast computer vision systems.

An interesting future development would be to expand the learning capabilities to other aspects of the model. An example is the adaptation of the weights controlling how much each agent's behaviour contributes to the overall behaviour. It would also be interesting to learn the behaviours themselves. An additional aspect that might be considered is the emergence of hierarchical organisation among the agents. Finally, the method's ability to deal with strong camera motion must be evaluated on a physical robot embodiment.

## Acknowledgements

This work was partially supported by FCT/MCTES grant No. SFRH/BD/27305/2006 and CTS multi-annual funding, through the PIDDAC Program funds.

## References

- Amari, S. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27(2):77–87.
- Bajcsy, R. (1988). Active perception. *Proceedings of the IEEE*, 76(8):996–1005.
- Ballard, D. H. (1991). Animate vision. *Artificial Intelligence*, 48(1):57–86.
- Beer, R. D. (2003). The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, 11(4):209–243.
- Couzin, I. (2009). Collective cognition in animal groups. *Trends in Cognitive Sciences*, 13(1):36–43.
- Doran, M. M., Hoffman, J. E., and Scholl, B. J. (2009). The role of eye fixations in concentration and amplification effects during multiple object tracking. *Visual Cognition*, 17(4):574–597.
- Floreano, D., Toshifumi, K., Marocco, D., and Sauser, E. (2004). Coevolution of active vision and feature selection. *Biological Cybernetics*, 90(3):218–228.
- Grassé, P.-P. (1959). La reconstruction du nid et les coordinations inter-individuelles chez *bellicositermes* et *cubitermes* sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux*, 6:41–80.
- Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259.
- Marshall, J. A. R. and Franks, N. R. (2009). Colony-level cognition. *Current Biology*, 19(10):395–396.
- Mirolli, M., Ferrauto, T., and Nolfi, S. (2010). Categorisation through evidence accumulation in an active vision system. *Connection Science*, 22:331–354.
- Neider, M. B. and Zelinsky, G. J. (2006). Scene context guides eye movements during visual search. *Vision Research*, 46(5):614–621.
- Nolfi, S. and Marocco, D. (2002). Active perception: a sensorimotor account of object categorization. In *Proceedings of the 7th International Conference on Simulation of Adaptive Behavior (SAB)*, pages 266–271. MIT Press.
- Oliva, A. and Torralba, A. (2007). The role of context in object recognition. *Trends in Cognitive Sciences*, 11(12):520–527.
- Passino, K. M., Seeley, T. D., and Visscher, P. K. (2008). Swarm cognition in honey bees. *Behavioral Ecology and Sociobiology*, 62(3):401–414.
- Rougier, N. and Vitay, J. (2006). Emergence of attention within a neural population. *Neural Networks*, 19(5):573–581.
- Santana, P., Alves, N., Correia, L., and Barata, J. (2010). Swarm-based visual saliency for trail detection. In *Proceedings of the IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems (IROS)*, pages 759–765. IEEE Press, Piscataway.
- Santana, P. and Correia, L. (2010). A swarm cognition realization of attention, action selection and spatial memory. *Adaptive Behavior*, 18(5):428–447.
- Santana, P. and Correia, L. (2011). Swarm cognition on off-road autonomous robots. *Swarm Intelligence*, 5(1):45–72.
- Scheier, C., Pfeifer, R., and Kuniyoshi, Y. (1998). Embedded neural networks: exploiting constraints. *Neural Networks*, 11:1551–1596.
- Sporns, O. and Lungarella, M. (2006). Evolving coordinated behavior by maximizing information structure. In *Proceedings of ALife X*, pages 3–7. The MIT Press, Cambridge, MA.
- Treisman, A. M. and Gelade, G. (1980). A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136.
- Trianni, V. and Tuci, E. (2011). Swarm cognition and artificial life. In *Proceedings of the 10th European Conference on Artificial Life (ECAL 2009)*, volume LNCS/LNAI 5777, 5778. Springer-Verlag, Berlin, Germany.
- Trianni, V., Tuci, E., Passino, K., and Marshall, J. (2011). Swarm cognition: an interdisciplinary approach to the study of self-organising biological collectives. *Swarm Intelligence*, 5(1):3–18.
- Tsotsos, J. K., Culhane, S. M., Kei Wai, W. Y., Lai, Y., Davis, N., and Nuflo, F. (1995). Modeling visual attention via selective tuning. *Artificial intelligence*, 78(1-2):507–545.
- Turner, J. (2011). Termites as models of swarm cognition. *Swarm Intelligence*, 5(1):19–43.
- Wolfe, J. M., Võ, M. L.-H., Evans, K. K., and Greene, M. R. (In Press). Visual search in scenes involves selective and nonselective pathways. *Trends in cognitive sciences*, doi:10.1016/j.tics.2010.12.001.
- Zhang, X., Hu, W., Maybank, S., Li, X., and Zhu, M. (2008). Sequential particle swarm optimization for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE Computer Society, Washington, DC.

# Co-evolution of pre-play signaling and cooperation

Francisco C. Santos<sup>1,2</sup>, Jorge M. Pacheco<sup>2,3</sup> and Brian Skyrms<sup>4</sup>

<sup>1</sup> Centro de Inteligência Artificial & Departamento de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal.

<sup>2</sup> ATP-group, CMAF, Complexo Interdisciplinar, P-1649-003 Lisboa Codex, Portugal.

<sup>3</sup> Departamento de Matemática e Aplicações, Universidade do Minho, 4710 - 057 Braga, Portugal.

<sup>4</sup> Logic and Philosophy of Science, School of Social Sciences, University of California at Irvine, Irvine, CA 92612, U.S.A.

email: fcsantos@fct.unl.pt

## Abstract

Understanding the evolutionary mechanisms that promote and maintain cooperative behavior is recognized as a major theoretical problem where the intricacy increases with complexity of the participating individuals. Costless pre-play communication [1] with signals that have no preexisting meaning (also known as cheap-talk) might not, on the face of it, be expected to do much. With the current extended abstract, here we would like to present a new analysis of this problem. This analysis has been recently reported in [Santos, F.C., Pacheco, J.M., Skyrms, B.: *Co-evolution of pre-play signaling and cooperation*. *J Theor Biol* 274 (2011) 30-35] [2].

Here, we show how pre-play signaling leads to profound changes in the evolutionary dynamics of cooperative games, favoring cooperation in finite populations. Cooperation freely emerges from the co-evolution of signals, assigned meanings and actions which are not built-in in the individual, addressing in a general framework the study of central aspects of Human evolution, from the self-organized drive towards an individual adoption of a given signaling system to the emergence of the latter [1].

We analyze two important metaphors of cooperation: The Stag-Hunt (SH) (or coordination) game and the Prisoner's dilemma (PD). We show how, on coordination dilemmas, individuals willing to cooperate learn how to use the information encoded in each signal to identify other cooperators, reducing the risk of facing defection upon a cooperative act. In addition, the existence of a large number of signals enhances the tendency to cooperate, as it enlarges the portfolio of available signals that cooperators may use at profit to coordinate. Since mutual cooperation is always the best possible outcome in coordination dilemmas, cooperators who are able to discriminate between their own strategy and the one of others are robust against the invasion of mutants. Consequently, the emergence of evolutionary stable strategies (and signals) requires that these strategies are *i*) cooperative, *ii*) discriminative and *iii*) self-reinforcing, that is, they cooperate with individuals who adopt the same signal.

Remarkably, the enhancement of cooperation through signaling also applies to games where deception constitutes a profitable option, and where defection is the only stable strategy, as in the PD. In the presence of pre-play signaling, those strategies that opt invariably to defect are no longer stable in the PD. However, the same remains true for any type of cooperative strategy. Let us suppose that mutant arises who can utilize an unused signal. The mutant sends the signal, cooperates with others who send it, and defects against the natives - who do not send it. All goes well for the invaders until another mutant arises who sends the signal and then defects. Thus, in the absence of any evolutionary stable strategy, the fate of cooperation emerges from the conflict between deception by fake signaling and development of reliable "secret handshakes" [3].

Finally, all results are shown to be strongly dependent on the number of signals available. In particular, cooperation can emerge as a result of the arms race between *i*) the exploration of new signals by cooperators (to avoid being cheated by defectors) and *ii*) the search of cooperative signals by defectors (to deceive cooperators). By increasing the number of signals, cooperators have a larger portfolio of signals to pick from, something they learn to use to their own advantage. This result illustrates the advantages of a complex signaling system (or incipient language system). Language, even if minimal may open a route to cooperation. Indeed, signaling systems, together with a rich communication portfolio, may give rise to a developing mechanism of intention recognition, from which future behaviors may be assessed and trust bonds established.

## References

1. Skyrms, B.: Signals: Evolution, Learning & Information. Oxford University Press (2010)
2. Santos, F.C., Pacheco, J.M., Skyrms, B.: Co-evolution of pre-play signaling and cooperation. *J Theor Biol* 274 (2011) 30-35
3. Robson, A.: Efficiency in Evolutionary Games: Darwin, Nash, and the Secret Handshake. *J Theo Bio* 144 (1990) 379-396

# An Experimental and Computational Approach to the Dynamic Body Boundary Problem

Yuki Sato<sup>1</sup>, Hiroyuki Iizuka<sup>2</sup> and Takashi Ikegami<sup>1</sup>

<sup>1</sup> General Systems Sciences, The Graduate School of Arts and Sciences, The University of Tokyo, 3-8-1 Komaba, Meguro-ku, Tokyo 153-8902, Japan

<sup>2</sup> Graduate School of Information Science and Technology, Osaka University, 2-1 Yamadaoka, Suita, Osaka 565-0871, Japan

yuki@sacral.c.u-tokyo.ac.jp

## Abstract

In this paper, we propose an experimental and computational model to challenge the dynamic body boundary problem, as seen in the rubber hands illusion and phantom limbs. Our strategy examines an agent's "attention shift". A computational model (Iizuka & Ikegami, 2005) was used to explore how a body and sensor can be made inseparable. A model agent was required to determine the number of vanes on a windmill by touching the windmill blindly with a stick. By adding an additional windmill to the first one, we investigated the agent's shift of attention, i.e. the agent could either determine the vanes on the first windmill, or the second windmill by using the first one as a tool. In other words, an agent's body image can shift from its arm tip to the boundary between the first and second windmill. We then introduced an experiment with a real windmill model to test the hypothesis shown by the theoretical model. Subjects were tasked with determining the number of vanes on the second windmill. We found that sensory-motor correlations between their actions and perceptions of the movement of the windmills were helpful for the attention shift but still not enough to extend their body boundaries.

## Introduction

### A Model to Bridge the Gap between the Self and the Environment

In order to overcome philosophical and scientific problems such as the "hard problem", which asks how and why certain neural processes give rise to subjective experiences (Chalmers 1996); or the symbol grounding problem, which asks how symbols get their meaning (Harnad 1990), we need a radical new framework or model to recast the dichotomies of mind and body, subject and object, agent and the environment, and perception and action.

How we model our cognition is directly connected to how we understand it. Studies on embodied robots and simulations are based on sensory-motor ideas that attempt to describe our psychological processes from sensory-motor connections and interactions with the environment (Walter 1950, 1951; Braitenberg 1984; Pfeifer & Scheier 1999; Brooks 1991a, b). For example, Walter (1950, 1951) discusses cognitive, play-like, and social behaviors by synthesizing artificial vehicles, while Braitenberg (1984) made conceptual robots to discuss the higher functioning of cognition. However, even if the above approaches succeed in shedding light on sensory motor experiences through interaction with the environment, the approaches still fails to consider psychological concepts such as body image, ownership, agency and active perception, which play an important role in resolving the dichotomies.

There are many phenomena in empirical neuropsychological studies that can be described with these psychological concepts. Yamamoto and Kitazawa (2001), for example, demonstrated in the arm-crossing experiment that the perceived temporal ordering of haptic stimuli was reversed when the successive stimuli were temporally close enough. Maravita and Iriki (2004) demonstrated that a macaque monkey's body image was extended to the tip of a tool bar when the monkey learns to use it. Ramachandran and Blakeslee (1998) showed that a human body image can be easily created or destroyed by using visual or auditory information. These experiments and others have revealed that body images and ownership have very dynamic natures, something we would like to implement in our system.

Our body image and ownership bridge the gap between the highly abstract sense of "self" and the physical world where our body is situated. Francisco Varela (1979) proposed a principle of autonomy, stressing the idea of a self-generated boundary. He exemplified autonomy as a "self" that emerged from a chemical system through structural coupling with the environment. In his model, it was shown that some reactive particles created a boundary, which regulated internal reactions of the particles, thus maintaining the boundary structure. The circularity of the physical boundary and the internal dynamics

produce the coherence of the self-state. In other words, the self has not been strictly defined but can be described as a dynamic process, and the sensory-motor experiences from the perspective of the emergent self, account for the psychological or highly abstract concepts such as life. One such challenge, with respect to a proto-cell system, can be seen in Suzuki and Ikegami (2004).

Mere sensory-motor modeling surrenders the self because it is pre-defined as a completely different entity to the environment and the boundary is given as the firm distinction between them. Therefore, we provide a new framework for modeling in order to achieve a balance between both ideas of emergent self and sensory-motor flow. In the new framework, we assume no explicit distinction between a sensor and a motor that defines the boundary. An interface between an agent and its environment is only dynamically constructed. Exploiting the model, we investigate active perception and body image as dynamic processes in the emergent self. The psychological notions are clarified first in this paper, after which the computational modeling and results are described. We also report some tentative results of a real windmill model, which has recently been made to investigate how human subjects feel their body boundaries.

## Body Image

Our body images are not restricted to the physical boundary that separates our bodies from the external world. When an expert driver drives a car, he/she can traverse narrow streets easily, as though the car were part of his/her own body. Indeed, he/she is aware of the whole car, and when the car runs over a rock, he/she feels as though he/she has stepped on it with his/her foot. Another example is an artificial tooth. We feel uncomfortable and cannot taste food when using an artificial tooth for the first time. However, over time, we adapt to the artificial tooth and learn to taste again. Yet another instance of this can be seen in a blind person's stick. As he/she adapts to its use, the stick changes from mere matter to a real body part, and the person is eventually able to perceive his/her environment with the stick. These examples show that one's body image can be extended beyond his/her physical body boundaries. Body images are formed through interactions between brain, body, tool, and environment. Nevertheless, the dynamic mechanisms underlying the changes of body images are still not fully understood, despite their importance in areas such as medical care, robotics, cognitive development, enactive cognitive science (Varela et al. 1991; Thompson 2007), the "extended mind" (Clark et al. 1998), and "radical embodied cognitive science" (Chemero 2009). We propose a model for body images by extending the windmill model proposed by Iizuka and Ikegami (Iizuka & Ikegami 2005). The windmill model proposed by Iizuka and Ikegami is a computer simulation model to study "active perception" (Gibson 1962).

## Active Perception

A difference between human perception and an artificial system based on current technology is the fact that two modes of perception exist in humans, active and passive modes of perception. When we touch an object with our hands, we perceive the shape, texture, and temperature of the object.

Gibson (1962) reported on experiments in which blind subjects touched different shapes of cookie cutters. When the cutter was pushed randomly on the subject's palm, the subject recognized the correct shape with 72% accuracy. By touching the cutter in a self-guided manner, the subjects recognized the object more than 95% of the cases. The former case is an example of passive perception and the latter case is active perception. This study illustrates that perception does not merely entail receiving information from the outside. It is instead a form of exploration. Moving our hands is not just a method we use to arrive at perception, but rather, the moving of one's hands is an ongoing exploratory process, which we think of as a generic property of perception. Edward Reed (1996) has further developed Gibson's idea, and this idea of perception has become a core principle of new psychology (often called ecological psychology).

Even though the idea of active and passive perception is subjectively apparent and has been studied empirically, it is still difficult to implement the two modes within the context of an artificial model. Iizuka and Ikegami (2005) studied the simulation model of object discrimination, which implement the two modes of perception. The present study further develops this research by studying the changing of body images.

## Computational Model

### Windmill Model for Active/Passive Perception

In this section, we briefly introduce the model for active/passive perception, and in the next section, we propose a model for body images and report results. In the proposed model, the agent consists of a body with a straight arm that can move and touch an object (Figure 1). The object is a windmill with a certain number of vanes, and the agent can rotate the windmill by pushing the vanes. This is what we call an active condition. When the agent perceives the windmill by its arm being pushed by the vane, this is a passive condition. In other words, the windmill has an infinite mass in the passive condition, and the agent cannot change the initial velocity of the windmill by pushing its vane. One of the aims of this windmill model was to examine the difference between the two methods of perception in terms of dynamic systems.

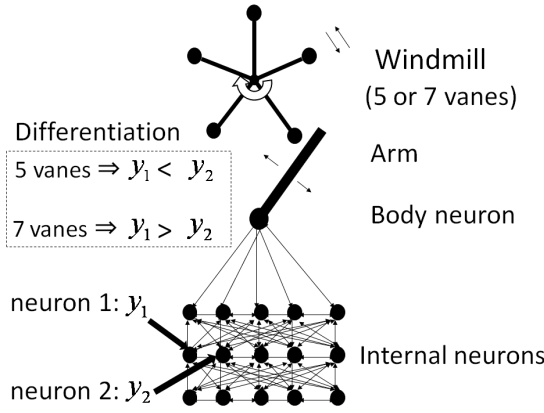


Figure 1: Windmill model for active/passive perception. The agent consists of an arm, a “body neuron”, and internal neurons. The windmill has 5 or 7 vanes. Differentiation of the windmills is made by comparing the activities of neurons 1 and 2.

Firstly, the dynamics of the arm and the windmill are controlled by the deterministic equation:

$$M_a \ddot{\theta}_a + D_a \dot{\theta}_a + F_a + F_{col}(\theta_a, \theta_w) = 0, \quad (1)$$

$$M_w \ddot{\theta}_w + D_w \dot{\theta}_w + F_{col}(\theta_a, \theta_w) = 0, \quad (2)$$

where  $\theta_a$  and  $\theta_w$  denote the angles of the arm and the windmill, respectively;  $M_a$  and  $M_w$  denote the mass of the arm and the windmill, respectively;  $D_a$  and  $D_w$  denote the friction coefficient of the arm and the windmill, respectively;  $F_{col}$  is a function giving the potential of collision; and  $F_a$  is the force of the agent used to rotate the arm.

Secondly, this agent also has a “brain” that consists of internal neurons (Figure 1). The dynamics of these neurons are controlled by a continuous-time recurrent neural network (CTRNN) (Beer 1995). The dynamics of the neural system are expressed by the following equations:

$$\tau_i \dot{y}_i = -y_i + \sum_{j=1}^M w_{ji} g_j(y_j), \quad (3)$$

$$g_i(x) = \frac{1}{1 + e^{-x - b_i}}, \quad (4)$$

where  $y$  is the state of each neuron,  $\tau$  is its time constant,  $b$  is a bias term, and  $w_{ji}$  is the strength of the connection from the neuron,  $j$  to  $i$ . It should be noted that we adopted a sparse structure rather than a fully-connected network. The neurons are arranged in 3 layers.

Thirdly, the agent has a body neuron at the interface between the arm and the internal neurons. The body neuron simultaneously plays the role of sensor and motor. That is, this neuron determines the value of  $F_a$  and the angle of the arm is assigned to the body neuron (Figure 2). The agent has no explicit functional division of sensors and motors. The distinction between moving and being moved becomes implicit. Whether an arm motion is caused spontaneously or

externally, it is internally evaluated by investigating the body neurons and internal neurons. In an empty space, an agent can freely move his/her arm. When an arm hits an object, the collision produces de-coherence of the arm movement, which is interpreted as sensory information.

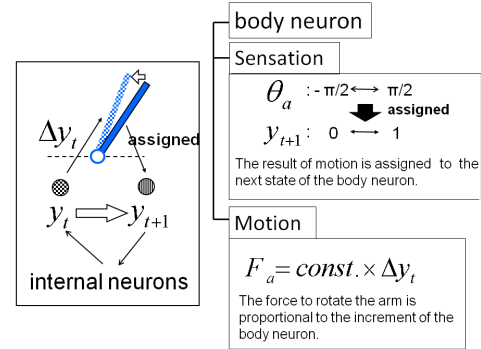


Figure 2: Updating the state of the body neuron, which plays the role of sensor and motor at the same time.  $\Delta y_t$  denotes the increment of the body neuron, which is given by equation (3).  $\Delta y_t$  is used to determine the force to rotate the arm.

**Active/Passive Agents.** An agent interacts with the windmill and distinguishes the number of vanes present (5 or 7) given the two conditions (cf. the beginning of the previous section). Specifically, an active agent distinguishes a windmill by actively touching the vanes. A passive agent does the same task by being pushed by the windmill. In both cases, this differentiation is made by comparing the neural activities of two neurons, neuron 1 and neuron 2 (Figure 1). If  $y_1$  is greater than that of  $y_2$ , the agent distinguishes the windmill as having 7 vanes, and if  $y_1$  is less than  $y_2$ , the agent distinguishes the windmill as having 5 vanes.

To train both active and passive agents, we adopted a standard genetic algorithm (GA) (Holland 1975) by encoding  $w_{ji}$  (the neural weight),  $\tau_i$  (time constant), and  $b_i$  (bias neural states) (cf. equation (3), (4)) into the real-valued strings. These strings are taken as artificial genomes and will be selected according to the fitness value of the corresponding agent (Figure 3). The value is calculated by multiplying the percentage of correct answers. The best-performing agent is preserved in the population without a genetic operation (elitism). The other agents are reproduced from the best agents by adding small random values (without sexual reproduction).

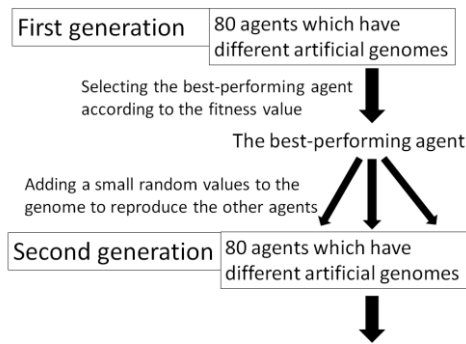


Figure 3: A schematic view of the genetic algorithm (GA) used in this study. We prepared 80 agents with different artificial genomes. The best-performing agent is selected according to the fitness value of the corresponding agent. The best-performing agent is preserved in the population. The other agents are reproduced from the best agents by adding small random values.

**General Observations.** A computational model shows that an agent becomes sensitive to the number of vanes. One difference exists in active and passive classifications: active classification is less stable against time delay compared to passive classification (Iizuka & Ikegami 2005). This is the dynamic system's interpretation of active and passive perception. In the following sections we further extend this model by adding the second windmill next to the first and gear the two windmills to move associatively. Our concern is to study how an agent's discrimination capability can be extended to the second windmill. We shall also discuss the synthesis of body images with the windmill.

### Coupled Windmill Model

In studying the coupled windmill model, we fix the number of vanes of the first windmill to 5 and require the agent to determine the number of vanes of the second windmill (which has 5 or 7 vanes). See Figure 4 for an illustration. An agent now uses the first windmill as a "tool" to determine the number of vanes on the second windmill. If an agent can successfully use the first windmill as a tool, we can say that, for the agent, the first windmill has shifted from an object to a tool. At this time, the agent's body image is thought to be extended to the first windmill. In the following sections, we only focus on the "active" agents, which actively use their arms to rotate the windmill in order to classify the number of vanes present.

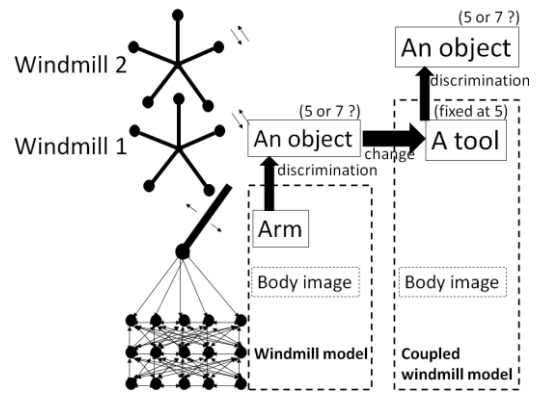


Figure 4: A coupled windmill model for studying body images. This agent determines the number of vanes on Windmill 2. In the previous windmill model, Windmill 1 was an object to be distinguished. On the other hand, in the coupled windmill model, Windmill 1 changed from an object to be distinguished to a tool to determine the number of vanes on Windmill 2. At this time, the agent's body image is thought to be extended to Windmill 1.

**Is the First Windmill a Tool or a Mere Object?** The first windmill is proposed to become an extension of the agent's body, thereby shifting his/her body image. If an agent can judge the number of vanes of the second windmill, can we identify this as an emergence of the body image? In this case, even if the agent can distinguish between the windmills, we cannot simply say that the agent has shifted his/her body image. The agent might just distinguish two windmills as  $(5, 5)^1$  and  $(5, 7)$ . In other words, the first windmill might not be a tool but a mere object. We cannot decide which is right if the agent differentiates between two combinations of windmills. These are,  $(5, 5)$ ,  $(5, 7)$ .

To overcome this problem, we required the agent to distinguish not two combinations  $((5, 5), (5, 7))$ , but four combinations, which are,  $(5, 5)$ ,  $(5, 7)$ ,  $(7, 5)$ , and  $(7, 7)$ . We want to compare two different agents to discuss the boundaries of body images. If an agent classifies the four combinations as two groups, which are,  $\{(7, 5), (7, 7)\}$  and  $\{(5, 5), (5, 7)\}$ , then the agent is sensitive to the vanes of the first windmill (this is called Agent 1) (Figure 5). This is because Agent 1 classifies the combinations within the same/different category if the first windmill has the same/different number of vanes and the agent does not care about the second windmill. In contrast, if an agent classifies the combinations as  $\{(5, 5), (7, 5)\}$  and  $\{(5, 7), (7, 7)\}$ , the agent is sensitive to the vanes of the second windmill (this is called Agent 2) (Figure 6). Here, Agent 2 classifies the combinations with respect to the second windmill and does not care about the first windmill. In other words, for Agent 1, the first windmill functions as an object to be distinguished, and the second windmill works as a noise (Figure 5). Or, we might say that the first windmill is perceived as a "figure" by Agent 1 and the second windmill is seen as a "ground". In contrast, for Agent 2, the second windmill becomes an observed object (or a "figure"), and the first windmill is a tool (or a "ground") to distinguish the second windmill (Figure 6).

<sup>1</sup> The first and second components  $( , )$  are the number of vanes of the first and second windmills, respectively.



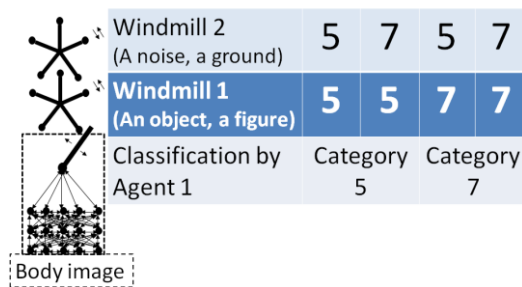


Figure 5: Agent 1. This agent is sensitive to the first windmill in the classification and does not care about the second windmill. The first windmill functions as an object to be determined, and the second windmill works as a noise, which means that the first windmill is not a part of the body image of Agent 1.

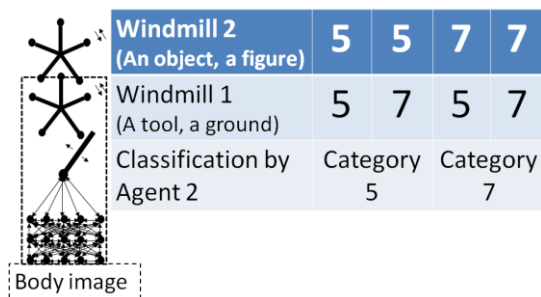


Figure 6: Agent 2. This agent is sensitive to the second windmill in the classification and does not care about the first windmill. The second windmill becomes an observed object, and the first windmill is a tool to distinguish the second windmill, which means that the first windmill is a part of the body image of Agent 2.

**Shift of Attention.** We think that this “shift of attention” is essential in defining the boundary of body image. For example, when we use a word processor for the first time, we pay attention not to the characters on the screen, but to the keyboard. At this stage, the keyboard is still an observed object and our body image is not extended to the keyboard. However, the attention is shifted from the keyboard to the screen as we become accustomed to typing. At this time, the keyboard’s status changes from being a mere object to a real tool, and our body image is extended to the keyboard. In our model, Agent 1 pays attention to the first windmill and does not care about the second windmill, which means that the first windmill is not part of the body image of Agent 1. In contrast, Agent 2 pays attention to the second windmill and does not care about the first windmill, which means that the first windmill is a part of the body image of Agent 2.

**Key Observations** By using a genetic algorithm, we successfully trained agents to become sensitive to the vanes of the first (Agent 1) or of the second windmill (Agent 2).

Changing the number of vanes successively from (5, 5) to (5, 7) to (7, 7) to (7, 5), we see that in the case of Agent 1, neuron 1 and neuron 2 are sensitive to the first windmill and do not care about the second windmill (Figure 7); in the case of Agent 2, neuron 1 and neuron 2 are sensitive to the second windmill and do not care about the first windmill (Figure 8).

For example, from (5, 5) to (5, 7), as in the case of Agent 1, no transition occurs in the neural states. However, in the case of Agent 2, a sharp transition occurs, and the magnitude relation is changed. In contrast, from (5, 7) to (7, 7) in the case of Agent 1, a sharp transition occurs, but in the case of Agent 2, the neural states maintain the magnitude relation.

As far as we know if we change the number of vanes, it won’t give the same result. However, as we already reported, a system properly count the number of the vane, when there is only one wheel (Sato et al. 2009). But counting two wheels case was also tough for the computational model.

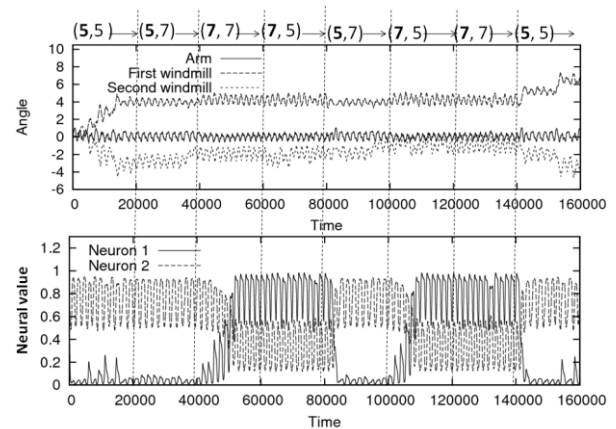


Figure 7: Agent 1: the time series of the arm, the first windmill, the second windmill (top) and of neurons 1 and 2 (bottom).

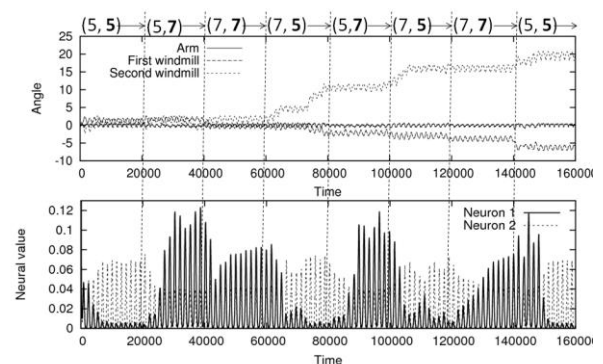


Figure 8: Agent 2: the time series of the arm, the first windmill, the second windmill (top) and of neurons 1 and 2 (bottom).

## Experimental Setup

### Real Windmill Model

In order to test the hypothesis shown by the theoretical model, we have conducted a real experiment and constructed two windmills with crossed metal bars. In this setup we fix the number of vanes on the first windmill to 5 and ask subjects to determine the number of vanes on the second windmill (which has 5 or 6 vanes).

Subjects wear a blindfold and touch the first windmill with only a stick, which is also fixed in space. The stick is introduced to constrain the movement of the subjects. Subjects are requested to determine the number of vanes on the second windmill in 30 seconds. The experiment is repeated over 30 trials.

**Result.** Subjects ( $N=5$ ) come to discriminate windmills about 80 percent accuracy at the end (Fig. 9a). Observationally, in the early stages, the stick and the windmills move randomly but they switch to regulatory behavior in the end in cases of a single (Fig. 9b) and coupled windmill experiment.

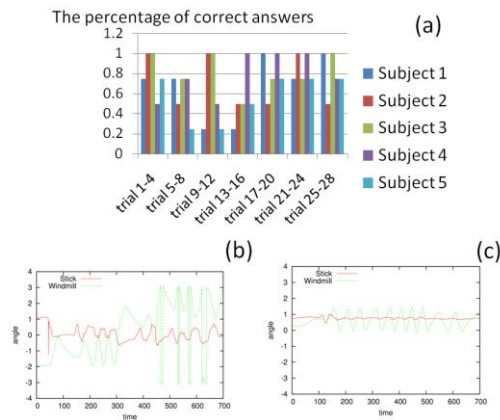


Figure 9; (a) The percentage of correct answers in a coupled windmill experiment. (b), (c) The time series of the stick and the windmill in a single windmill experiment. Movement of the stick changes from random motion (b) to periodic motion (c) as the subjects adapt to its use.

**The body boundary of subjects is still not extended.** Although subjects could count the number of vanes usually, they reported that they just paid attention to the touch feeling of collisions between the stick and the first windmill (Fig. 10a).

Because the number of vanes on the first windmill is fixed to 5, the collision events between the stick and the first windmill increase in frequency when the second windmill has 6 vanes, and decreases if it has fewer vanes ( $\neq 5$ ). With this trick, subjects could count the vanes on the second windmill. In this case, the first windmill is still an object to the subjects so that the body boundary is not extended to the boundary between the first and second windmill.

Figure 10b shows the time series of the positions of the stick and the vane of the first windmill which collide with the second windmill (the red vane in Fig. 10a). The supporting point of

the stick and the windmills are fixed on the horizontal line in Fig. 10a. But the center of the oscillation of the stick and the vane of the first windmill is not on the line (Fig. 10b).

Something is needed to extend a subject's body boundary. Our hypothesis is that subjects need more visual information about the windmills to learn the sensory-motor correlations between their action and the movement of the windmills. But if subjects can see the windmills they also recognize the number of vanes of the second windmill.

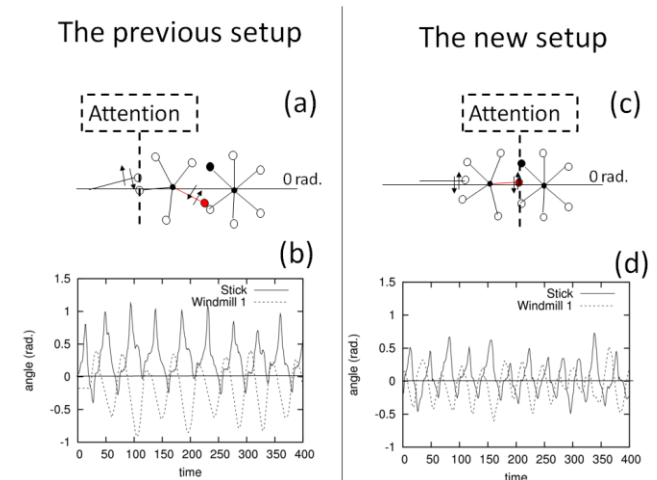


Figure 10: (a) The subjects reported that they paid attention to the collisions between the stick and the first windmill. The supporting point of the stick and the windmills are fixed on the horizontal line. (b) The direction of the first windmill from the supporting point of the stick is 0 radian (the horizontal line in Fig. 10a), but the center of the oscillation of the stick and the vane of the first windmill which collide with the second windmill (the red vane) is less than 0 radian. (c) In the new setup some subjects paid attention to the collisions between the first windmill and the second one and did not care about the collisions between the stick and the first windmill. The supporting point of the stick and the windmills are fixed on the horizontal line. (d) The direction of the first windmill from the supporting point of the stick is 0 radian (the horizontal line in Fig. 10c), and the center of the oscillation of the stick and the vane of the first windmill which collide with the second windmill (the red vane) is 0 radian.

### New Setup

In order to extend the body boundary of subjects, we introduced visual inputs (Fig. 11). A video camera captures the windmills and displays them on the monitor, while subjects do the task, watching the monitor.

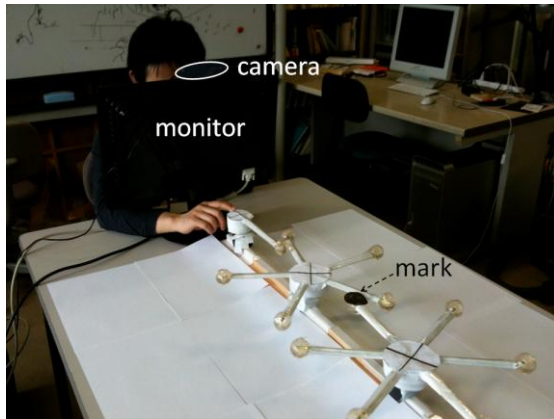


Figure 11: New setup. Here, there is a stick, the first windmill, and the second windmill, which is white. There is also a monitor, camera and one black mark attached to a vane on the second windmill. The camera captures the windmills and displays them on the monitor.

Now subjects can observe the global configuration of two windmills and how they move around (the left image of Figure 12). By using a black-white screen and painting the two windmills in different colors, a subject can only see the movement of one windmill at a time.

In the right image of Figure 11, subjects can only see the second windmill. Since we only put a mark on one vane, subjects can't recognize the number of vanes, but they can see the movement of the second windmill.



Figure 12: Two kinds of images on the monitor. The left image has full color. In this case subjects can see all and recognize the number of vanes. On the other hand, the right image is in black and white. In this case, subjects can only see the black mark attached to a vane of the second windmill.

### Observations of the new setup

In the 14th trial, a subject reported that he discovered how to use the first windmill to distinguish the number of vanes on the second windmill. He tried to use one vane on the first windmill (the red vane in Fig. 10c) to oscillate the second windmill. He reported that he felt as if the first windmill was the stick to distinguish the second one. He also reported that he paid attention to the collisions between the first windmill and

the second one and did not care about the collisions between the stick and the first windmill (Fig. 10c).

Figure 10d shows the time series of the stick and the vane on the first windmill which collides with the second windmill (the red vane) in the 14th trial. The center of the oscillation of the stick and the vane of the first windmill is on the horizontal line in Fig. 10c.

A remarkable difference between Fig. 10b and Fig. 10d is the following. When the visual information is available (Fig. 10d), subjects try to use a vane of the first windmill (the red vane) as a "controlling handle" to move the second windmill. As a result, that vane and the stick before the first windmill align in a straight line.

## Summary and Discussion

In this paper, we firstly demonstrated that even simple computational agents can have two different sensitivities to the windmills. It should be worth noting that the agents can ignore the number of vanes of the unattended windmill. An agent becomes either sensitive to the first windmill or the second one, neglecting the other. We claim that this shift of attention from the first windmill to the second is a dynamic shift of the body boundary.

In the real windmill model, we found that there are two ways to distinguish the second windmill. In the previous setup, subjects do the task with a blindfold. In this case subjects could not learn the sensory-motor correlations between their action and the movement of the windmills' vanes, and felt that the first windmill was an object to be distinguished. On the other hand, in the new setup subjects could see the movement of a vane on the second windmill, so some subjects could learn the sensory-motor correlations between their action and visual information of the second windmill. In this case, some found how to use the first windmill as a tool to distinguish the second windmill, and they could pay attention not to the collisions between the stick and the first windmill but to the collisions between the first windmill and the second one.

But this shift of attention is still weak and not enough to extend their body boundaries for most of the subjects. We are now planning to change the material of the ball attached to the tip of the vanes to a heavier material, so that subjects can feel the collisions between the first windmill and the second one clearly. It will help subjects with a blindfold to determine the movement of the second windmill and to learn the sensory-motor correlations between their action and perception. Some reported that due to the noisy setting up of the experiment, it was difficult to predict the movement which prevented the body boundary from extension. Also we are afraid that since the present setting up uses a single stick + a first windmill + second windmill, the discrimination task became inevitably complex. We are improving the point to simplify the structure.

The value of this paper lies in the ambiguity of the first windmill, which is a tool (a part of a subject) and an object (an environment) at the same time. Our insights are beneficial for the biology of cognition, enactive cognitive science, the "extended mind" (Clark et al. 1998), and "radical embodied cognitive science" (Chemero 2009; Dotov et al. 2010). In our

study, the dichotomy of object and subject is rejected and the active role of an observer in perception is considered.

We argue that the ambiguity of the first windmill corresponds to the ambiguity of our body, something that is known in German as *Körper* (a physical living body) and *Leib* (a subjectively lived body) (Thompson 2007: 231). The two aspects of our body are intimately related to changes of our body images. For example, a blind person's stick changes from a mere object (*Körper*) to a real hand (*Leib*) when he/she adapts to it. In our model, on the one hand, the first windmill is observed as a material thing in the world (by Agent 1), which means the first windmill is *Körper* at this time. On the other hand, the first windmill is used to perceive the second windmill (by Agent 2), which means the first windmill functions as *Leib* at this time.

From this point of view, we need to recast the "hard problem". Thompson recasts the explanatory gap between mental and physical as the *body-body problem*: the problem of relating one's subjectively lived body to the organism or living body that one is (Thompson 2007: 244).

Moreover, we are extending the current model to study communications between two agents by introducing one more agent instead of the second windmill. The two agents interact with each other through the first windmill and discriminate each other's neural state. The agents convey and receive messages. At this time the windmill functions as their interface or some kind of "language". Also, these agents eventually conform their neural states with each other. We think this is a kind of primitive communication (empathy or imitation).

In this way we could understand the course of humankind's mental development from active perception to extension of body images, and to inter-subjective communication by extending our windmill model further. We will also employ our model for robot learning by using a servo motor.

## References

- Beer, R. D. (1995) On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, vol13, no 4: 469-509.
- Bovinick, M. & Cohen, J. (1998) Rubber hands 'feel' touch that eye see. *Nature* 391: 756.
- Braitenberg, V. (1984). *Vehicles: Experiments in synthetic psychology*. MIT Press, Cambridge.
- Brooks, R. A. (1991a) Intelligence without representation. *Artificial Intelligence Journal* (47): 139-159.
- Brooks, R. A. (1991b) New approaches to robotics. *Science* 253: 1227-1232.
- Chalmers, D. (1996) *The conscious mind: In search of a fundamental theory*. Oxford University Press, USA.
- Chemero, A. (2009), *Radical embodied cognitive science*. MIT Press, Cambridge.
- Clark, A. & Chalmers, D. (1998) The extended mind. *Analysis* 58:10-23.
- Dotov, D., Nie, L., & Chemero, A. (2010) A demonstration of the transition from readiness-to-hand to unreadiness-to-hand. *PLoS ONE* 5(3): e9433.
- Gibson, J. (1962) Observation on active touch. *Psychological Review* 69:477-491.
- Harnad, S. (1990) The symbol grounding problem. *Physica D* 42: 335-346.
- Holland, J. H. (1975) *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor.
- Iizuka, H. & Ikegami, T. (2005) Emergence of body image and the dichotomy of sensory and motor activity. In proceedings of the symposium on Next Generation Approaches to Machine Consciousness: 104-109, Hatfield, UK.
- Iriki, A., Tanaka, M., & Iwamura, Y. (1996) Coding of modified body schema during tool use by macaque postcentral neurons. *Neuroreport* 7:2325-30.
- Pfeifer, R., & Scheier, C. (1999) *Understanding intelligence*. MIT Press, Cambridge.
- Reed, E. S. (1996) *Encountering the world: Toward an ecological psychology*. Oxford University Press, New York.
- Ramachandran, V. S. & Blakeslee, S. (1998) *Phantoms in the brain: Probing the mysteries of the human mind*. William Morrow, New York.
- Sato, Y., Iizuka, H., & Ikegami T., Generation of Body Images in an Active/Passive Perception Model, In, Proceedings of the Fifth Asia-Pacific Computing and Philosophy Conference, Tokyo, pp.100-105, (October 2009)
- Suzuki, K. & Ikegami, T. (2004) Self-repairing and mobility of a simple cell. *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*: 421-426. MIT Press.
- Thompson, E. (2007) *Mind in life: biology, phenomenology, and the sciences of mind*, Harvard University Press.
- Varela, F. (1979) *Principles of Biological Autonomy*, North-Holland
- Varela, F., Thompson, E. & Rosch, E. (1991) *The embodied mind: Cognitive science and human experience*, MIT Press, Cambridge.
- Walter, W. G. (1950) An imitation of life. *Scientific American* 182(5): 42-45.
- Walter, W. G. (1951) A machine that learns. *Scientific American* 185(2):60-63.
- Yamamoto, S. & Kitazawa, S. (2001) Reversal of subjective temporal order due to arm crossing. *Nature Neuroscience*, 4(7): 759-765.



# Quantifying Evolutionary Dynamics of Swarm Chemistry

Hiroki Sayama<sup>1</sup> and Chun Wong<sup>1</sup>

<sup>1</sup> Binghamton University, State University of New York  
sayama@binghamton.edu

This paper reports our recent efforts to quantitatively characterize the evolutionary dynamics of self-organizing patterns observed in Swarm Chemistry.

Swarm Chemistry (Sayama 2009) is an artificial chemistry framework that can demonstrate self-organization of dynamic patterns of kinetically interacting heterogeneous particles. A swarm population in Swarm Chemistry consists of a number of simple self-propelled particles moving in a two-dimensional continuous space. Each particle can perceive average positions and velocities of other particles within its local perception range, and change its velocity in discrete time steps according to kinetic rules similar to those of Reynolds' Boids (Reynolds 1987). Each particle is assigned with its own kinetic parameter settings (similar to genotype) that specify preferred speed, local perception range, and strength of each kinetic rule. Particles that share the same set of kinetic parameter settings are considered of the same type. Several model extensions introduced in our recent work, including local information transmission among particles and their stochastic differentiation/re-differentiation, have made the model capable of showing morphogenesis and self-repair (Sayama 2010) and autonomous ecological/evolutionary behaviors of self-organized "super-organisms" made of a number of swarming particles (Sayama 2011; see Fig. 1).

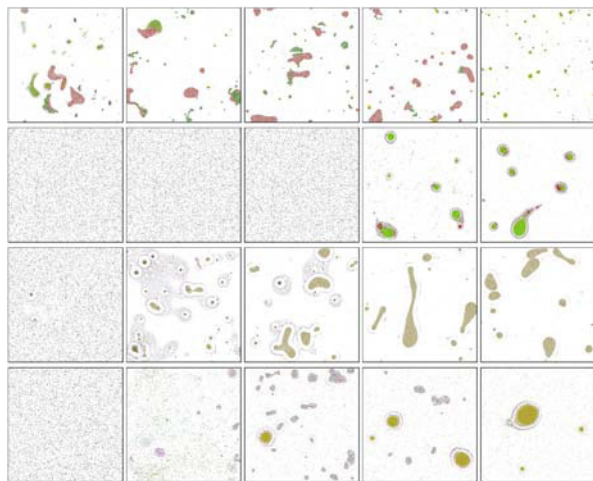


Figure 1. Typical evolutionary processes emerging in Evolutionary Swarm Chemistry (taken from (Sayama 2011)). Time flows from left to right. Four cases with different initial conditions are shown.

Our latest results (Sayama 2011) produced a hypothesis that the introduction of a high volume of mutations and dynamic exogenous perturbations helps a swarm population to break an established status quo and demonstrate more continuous evo-

lutionary exploration. However, the experimental results were evaluated so far by visual inspection only, with no objective measurements involved, and hence the hypothesis was not tested in a quantitative way.

To address the lack of quantitative measurements, we developed and tested two simple measurements to quantify the degrees of evolutionary exploration and macroscopic structuredness of swarm populations. These measurements were designed so that they can be easily calculated *a posteriori* from a sequence of snapshots (bitmap images) taken in past simulation runs, without requiring genotypic or genealogical information that was typically assumed available in other proposed metrics (Bedau and Packard 1992; Bedau and Brown 1999; Nehaniv 2000).

Evolutionary exploration was quantified by counting the number of new RGB colors that appeared in a bitmap image of the simulation snapshot at a specific time point for the first time during each simulation run. Since different particle types are visualized with different colors in Swarm Chemistry, this measurement roughly represents how many new particle types emerged during the last time segment.

Macroscopic structuredness was quantified by measuring a Kullback-Leibler divergence (Kullback & Leibler 1951) of a pairwise particle distance distribution from that of a theoretical case where particles are randomly and homogeneously spread over the entire space. Specifically, each snapshot bitmap image was first analyzed and converted into a list of coordinates (each representing the position of a particle, or a colored pixel), then a pair of coordinates were randomly sampled from the list 100,000 times to generate an approximate pairwise particle distance distribution in the bitmap image. The Kullback-Leibler divergence of the approximate distance distribution from the homogeneous case is larger when the swarm is distributed in a less homogeneous manner, forming macroscopic structures.

We first applied these measurements to two experimental conditions studied before (Sayama 2011): one with low mutation rates and static environments, called "original-low", and the other with high mutation rates and dynamical exogenous perturbations, called "original-high". Results are summarized in Figs. 2, 3 and 4 (marked by circles and squares, respectively). Figure 2 clearly shows the high evolutionary exploration occurring in the "original-high" condition, supporting our hypothesis quantitatively (but the exploratory dynamics generally decline over time). However, Figure 3 shows a downside of the "original-high" condition that it tends to destroy macroscopic structures by allowing swarms to evolve toward simpler, homogeneous forms.

A possible reason for this degradation of structuredness over time was already indicated in (Sayama 2011). Namely, the previous implementation of collision detection in Swarm Chemistry mistakenly depended on perception ranges of particles, so if a perception range of a particle evolves close to

zero, its kinetic properties will no longer change through interaction with other particles, and therefore the near-zero perception range worked as an artificial genotypic attractor.

We fixed this problem by implementing a minor modification to the collision detection rule so that a non-zero collision distance is always maintained. We call these conditions “revised- $\ast$ ” (where  $\ast$  is either “low” or “high”). The effect of this modification on evolutionary dynamics was measured by running a new set of simulations and then applying the proposed measurements to them. Results are marked by diamonds and triangles in Figs. 2, 3 and 4, which quantitatively showed that the “revised-high” condition successfully maintained macroscopic structures at the minor cost of evolutionary exploration.

This work was supported in part by the Binghamton University EvoS Small Grant (FY 2011).

## References

- Sayama, H. (2009). Swarm Chemistry. *Artificial Life*, 15:105-114.
- Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics* 21(4):25-34.
- Sayama, H. (2010). Robust morphogenesis of robotic swarms. *IEEE Computational Intelligence Magazine*, 5(3):43-49.
- Sayama, H. (2011). Seeking open-ended evolution in Swarm Chemistry. In *Proceedings of the Third IEEE Symposium on Artificial Life (IEEE ALIFE 2011)*, IEEE, in press.
- Bedau M. A. and Packard, N. H. (1992). Measurement of evolutionary activity, teleology, and life. *Artificial Life II*, pp.431-461.
- Bedau, M. A. and Brown, C. T. (1999). Visualizing evolutionary activity of genotypes. *Artificial Life* 5:17-35.
- Nehaniv, C. L. (2000). Measuring evolvability as the rate of complexity increase. *Artificial Life VII Workshop Proceedings*, pp.55-57.
- Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79-86.

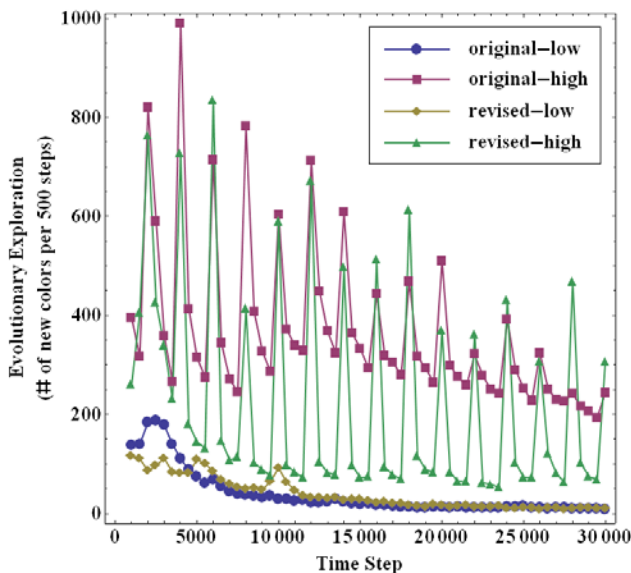


Figure 2. Temporal changes of the evolutionary exploration measurement (i.e., number of new colors per 500 time steps) for four different experimental conditions, calculated from snapshots of simulation runs taken at 500 time step intervals. Each curve shows the average result over 12 simulation runs (3 independent runs  $\times$  4 different initial conditions given in (Sayama 2011)). Sharp spikes seen in “high” conditions were due to dynamic exogenous perturbations.

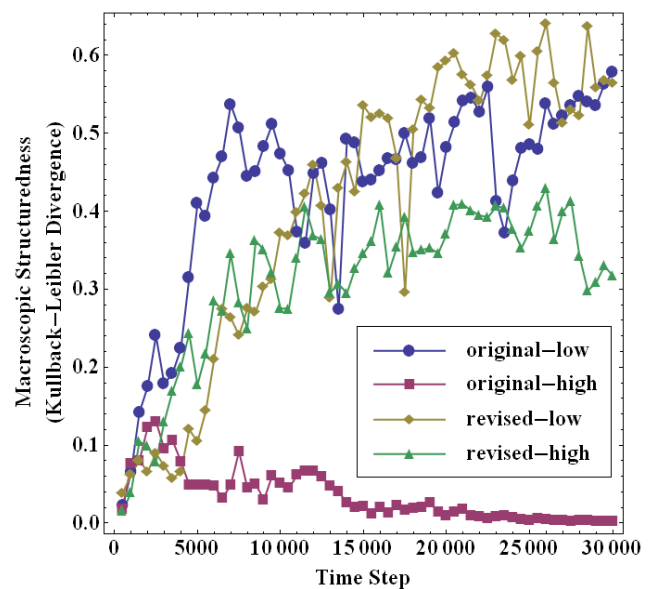


Figure 3. Temporal changes of the macroscopic structuredness measurement (i.e., Kullback-Leibler divergence of the pairwise particle distance distribution from that of a purely random case) for four different experimental conditions, calculated from snapshots of simulation runs taken at 500 time step intervals. Each curve shows the average result over 12 simulation runs (3 independent runs  $\times$  4 different initial conditions). The “original-high” condition loses macroscopic structures while other conditions successfully maintain them.

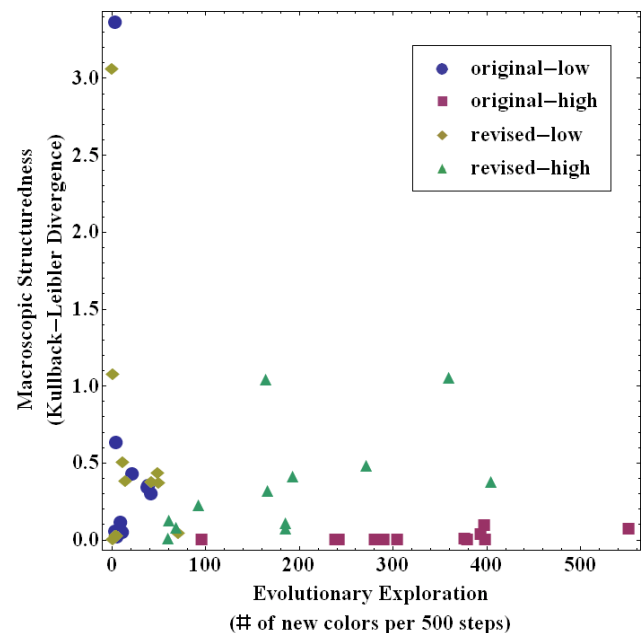


Figure 4. Evolutionary exploration and macroscopic structuredness averaged over  $t = 10,000 \sim 30,000$  for each independent simulation run. Each marker represents a data point taken from a single simulation run. It is clearly observed that the “revised-high” condition most successfully achieved high evolutionary exploration without losing macroscopic structuredness.

# Learning by seeing—associative learning of visual features through mental simulation of observed action

Malte Schilling<sup>1,2</sup>

<sup>1</sup>International Computer Science Institute Berkeley, CA 94704

<sup>2</sup>Center of Excellence 'Cognitive Interaction Technology', University of Bielefeld, D-33501 Bielefeld, Germany  
malteschilling@googlemail.com

## Abstract

Internal representations employed in cognitive tasks have to be embodied. The flexible use of such grounded models allows for higher-level function like planning ahead, cooperation and communication. But at the same time this flexibility presupposes that the utilized internal models are interrelating multiple modalities. In this article we present how an internal body model serving motor control tasks can be recruited for learning to recognize movements performed by another agent. We show that—as the movements are governed by an equal underlying internal model—it is sufficient to observe the other agent performing a series of movements and that there is no supervised learning necessary, i.e. the learning agent does not require access to the performing agents postural information (joint configurations). Instead, through the shared underlying dynamics the mapping can be bootstrapped by the observing agent from the sequence of visual input features.

## Introduction

Internal representation are essential in higher-level cognitive tasks. Following the view of embodied cognition internal models have to be grounded and are therefore nowadays assumed to be directly linked to the action-perception-cycle. Grounded models appear to be a byproduct which originally served a quite specific action and co-evolved in this context (Steels, 2003). But on a later-stage cognition has taken over and the same models could be applied in a more flexible way outside the original context of the grounding action. An example are targeted movements which can be found even in quite simple lifeforms as are insect. Nonetheless, making a targeted movement presupposes an internal model allowing to choose the correct muscle activation to reach a target which was perceived before in a three dimensional space. The ability to use this model not only in the context of one specific type of movements, but to also use the incorporated knowledge—i.e. how muscle activations and target positions are related—in a broader context appears to be essential for cognition. It is assumed that internal simulation is a key mechanism to recruit internal models for high level tasks (Hesslow, 2002). Planning ahead can be understood in this way as using only the internal representation decoupled

from the represented body to simulate behaviors and predict their consequences. This allows to try out possibly hazardous actions and to evaluate possible alternatives or slight modifications. Findings from diverse fields as neuroscience, psychology and behavioral sciences have contributed over the last years and shaped this view (Jeannerod, 2006). It is now more and more apparent that such a mechanism is at the core of cognition, but also subserves—and is grounded in—action and perception. Perception seems to be shaped by the encoded knowledge and when perceiving others performing actions it seems that internal models of the own body are used (Schacter et al., 2007). Perception tries to fit the perceived input to the representation grounded in motor control to make sense of what is perceived in the sense of the language of the own motor system. This becomes of course even more important in cooperation or communication as different roles between different subjects require different representations of what is going on.

A central issue is the multimodal nature of the underlying representational system. It appears that our conceptual system is—besides organizing concepts—binding diverse sets of features from different modalities and on different levels of abstraction. But how is it possible to come up with these connections and interrelate multiple modalities? In this paper, we want to address how associations between an internal body model used for motor control and visual representations can be established in an unsupervised way, i.e. only through observing another agent performing body movements an observing agent can come up with a mapping of the perceived visual features onto its own body model representation (segment orientations). As an example we use a three segmented arm and will first introduce a neural network which allows for motor control and makes targeted movement. In the third section we will explain how this body model can be incorporated into the perception loop and how it can subserve perception as it provides predictions of the movement and helps to disambiguate or filter noisy input. Even though only visual input is accessible when observing another agent performing movements, we afterwards show in principle that it is possible to come up



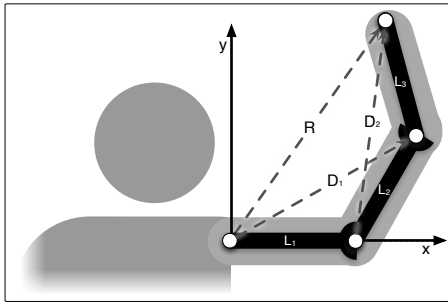


Figure 1: Graphic representation of a three segmented arm, consisting of upper arm ( $L_1$ ), lower arm ( $L_2$ ) and hand ( $L_3$ ). Vector  $R$  points to the position of the end effector (tip of the hand).  $D_1$  and  $D_2$  describe additional diagonal vectors. The arm is restricted to work in a plane (coronal plane).

with a mapping from the visual features to ones own motor system. The shared governing dynamics are enough to allow bootstrapping this mapping from the sequence of visual features. First results from computer simulations will be presented indicating that this mapping can be established quickly.

### MMC Networks as a Body Model

Mean of Multiple Computation networks are a type of recurrent neural networks (Cruse and Steinkühler, 1993; Steinkühler and Cruse, 1998). They are based on the Mean of Multiple Computation (MMC) principle which is allowing to use known constraints to set up the network instead of training the weights. In principle the constraints are given as equations which form the attractor space. The network is in this way similar to a self-organizing map as the constraints are enforced on any input given to the network. MMC networks have been used in the past for diverse kinematic tasks and we will use a simple example from this domain for explaining the principle. The general approach will be illustrated using a three segmented arm which can be moved around in a plane. The orientation of each segment is described as a two dimensional vector for illustrative purposes. A joint angle representation can be used as is applied usually in robotics and the approach has been extended to three dimensional movements (Schilling, 2011), for describing movement dynamics (Schilling, 2009) and a hierarchical organization in order to represent complex structures has been introduced (Schilling and Cruse, 2007). It is important to note that even though the task used here is quite simple, it is so complex that an analytical solution is not feasible as there are more degrees of freedom to be controlled as there are in the target space (Bernstein, 1967). The arm is redundant. Therefore, even in this simple example all the demanding characteristics are present we are facing in the control of complex movements, e.g. of a robotic or a human arm.

The manipulator is shown in fig. 1. Kinematic equations

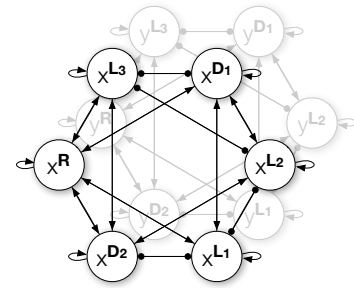


Figure 2: The MMC network consists of two identical networks, one for the x-components (black lines) and one for the y-components (grey lines) of the vectors. The units represent the components of the six vectors  $L_1$ ,  $L_2$ ,  $L_3$ ,  $D_1$ ,  $D_2$  and  $R$  of the planar arm. Connections with a positive weight are indicated by a black arrowhead and negative weights are shown as black dots. All connections are bidirectional. The example equation  $x^{D_1} = x^{L_1} + x^{L_2}$  is shown on the right: connections between the three nodes on the right encode all the equations derived, e.g.  $x^{L_1}$  is given through  $x^{D_1}$  and the negative value of  $x^{L_2}$ .

describing the arm can be easily set up. The main idea of the MMC approach is to not compile the kinematic relations into one single equation, e.g. representing the end position of the arm, but to establish a set of local relationships capturing the redundancy of the arm. As illustrated in the figure, additional diagonal vectors are introduced. A local relationship then corresponds to a triangle formed by three vectors, e.g. the first two segments and the first diagonal constitute such a triangle. As each triangle establishes a closed polygon chain, these relationships can be expressed as an equation, e.g. for the example above we will get  $x^{D_1} = x^{L_1} + x^{L_2}$ . (and an analogous equation for the y-component) Overall a set of equations can be compiled following this approach when all possible triangle relationships are constructed. Each vector variable is taking part in multiple of these equations. In the next step for setting up the network, for each variable all equations containing that variable are solved with respect to that variable. In our example, the first segment is contained in one additional equation. Solving these two equations for the first segment variable we get:

$$\begin{aligned} x^{L_1} &= x^R - x^{D_2} \\ x^{L_1} &= x^{D_1} - x^{L_2} \end{aligned} \quad (1)$$

Following the Mean of Multiple Computation principle these multiple computations for one variable are integrated through calculating the Mean value. In order to restrain that abrupt and fast changes in one equation affect the whole process, usually as an additional term the weighted old value is included into the mean computation which introduces a sort of damping (Makarov et al., 2008). As a result in our exam-

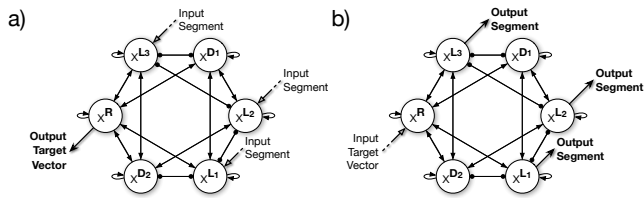


Figure 3: Application of the network to solve kinematic tasks. Initially, the network is in a stable state reflecting the current configuration of the arm (see Fig. 1). In a) it is shown how the net solves the forward kinematic task, i.e. when the segment orientations are known the end effector position can be computed. In b) the application for solving the inverse kinematic task is shown. A target position is given as input to the network and the network adjusts the segment vectors accordingly. If an input is given, the corresponding recurrent channels are suppressed (symbolised by the open arrow heads).

ple this leads to

$$x^{L_1}(t+1) = \frac{1}{d}(x^R(t) - x^{D_2}(t)) + \frac{1}{d}(x^{D_1}(t) - x^{L_2}(t)) + \frac{d-2}{d}x^{L_1}(t) \quad (2)$$

This set of equations describe the relations between the variables and can be understood as defining the connections of a neural network. The network is shown in fig. 2. As the resulting network is a recurrent neural net, the activation of the network is developing over time in which the state of the network can be calculated in an iterative fashion. The encoded constraints enforce this behaviour and the attractor space reflects states fulfilling all the kinematic equations. Obviously, when we give a valid configuration of the arm to the network all constraints are met and the network is in a stable state (Steinkühler and Cruse, 1998). The interesting cases are the cases in which we only provide partial information. Acting like a self-organising map the MMC net completes the given input pattern into a corresponding activation of the whole network which matches the requirements. In this way the net is able to solve any kinematic problem.

The forward kinematic problem can be solved straightforward. As an input the segment vectors are fed into the network (fig. 3 a). The corresponding diagonal and end-effector vectors are approached in a few time steps (depending on the damping factor, i.e. the weight of the recurrent connection). Importantly, the input to the network is given to the network the whole time and is directly setting the input variables.

For the inverse kinematic task, we only give as an input the desired end-effector position to the network (shown in fig. 3 b) after initialising the network with a valid starting

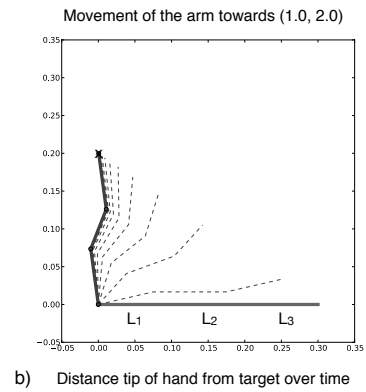


Figure 4: Solution of the inverse kinematic problem through the linear MMC model. A planar arm with three segments (i.e., one extra DoF) should point to a given position, marked by a cross, starting from an initial configuration. The state of the arm for every second iteration step is shown.

configuration. Through enforcing the new end effector value onto the network, a disturbance is introduced and the network is not in an attractor state anymore. But over time this activity is spread to all variables. The encoded kinematic constraints enforce that the network settles back on its solution space. The network relaxes to a stable state in which the target end effector value still holds true and the other variables have adopted corresponding values. As an example, we show in figure 4 an example run of the network. Initially, the arm is fully stretched to the right (bright line, end effector position  $x = 0.3, y = 0$ , with all segments having an equal length of 0.1 units). For every second iteration step the current configuration of the arm is shown (dashed grey line), until the 25th iteration in which the arm has reached the target position ( $x = 0, y = 0.2$ , drawn as a solid dark line).

As can be seen in this example—and as has been shown in the past (Steinkühler and Cruse, 1998)—the MMC network is able to solve the inverse kinematic task in only a few iteration steps. We presented the linear MMC network above which has one serious drawback as it allows the variables to change freely. There is no cross connection between the  $x$  and the  $y$  component of the networks. As the  $x$  and  $y$  components of the variables can be modified independently the length of the vectors can change. This is usually unwanted and problematic for the segment variables which should stay of constant length. This problem can be easily solved through a normalisation step. Even though this introduces non-linearities into our network this does not disrupt the overall performance (Steinkühler and Cruse, 1998). In this article we will apply such a normalisation step on the MMC segment variables (not shown in the diagrams) after each iteration step which could be totally circumvented when using other representations like a joint angle represen-

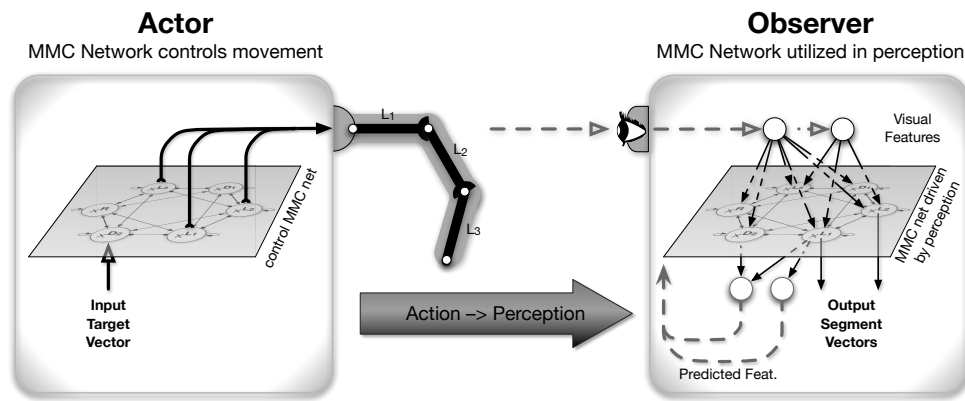


Figure 5: Application of the internal model in action (shown on the left) and perception (right). The internal model is used in one agent for motor control. Given a target vector as input it comes up with a movement to the target. On the other side, an equal internal model is utilized in another agent during perception. The embedded dynamics of movements allow to establish a connection from visual features to body postures and to recognize postures of another actor when seeing them.

tation(Schilling, 2011).

A property one immediately recognises when looking at the behaviour of the network is that the arm is moved in the beginning very fast and later-on dramatically slows down. The distance to the target decreases exponentially. Biological movements, e.g. human arm movements, are characterised by very different properties (Morasso, 1981). Again, we introduced MMC networks in the past which incorporate dynamic influences and which nicely fit to experimental data for human reaching movements (Schilling, 2009).

### Application of the Body Model in Perception

Internal models are used in motor control, e.g. in reaching tasks inverse models transform target points into joint positions or muscle activation. The introduced MMC network implements such an internal model of the own body and allows for making targeted movements. But the same internal models have been found active in other tasks, e.g. perception, planning ahead or communication (Grush, 2004). It appears that internal models are recruited by these other function. While in this way the utilized internal model is grounded in action, it remains unclear how it can be connected to seemingly quite different tasks. As we want to show in this paper, the underlying organisation of the body model is providing enough structure (in time) to allow for establishing such connections. We are focussing on the use of an internal body model in perception of movements. A key question is how humans and even simple animals are able to recognize and understand movements of conspecifics. It has been pointed out that mapping an observed behaviour to ones own body model is essential (Decety and Grezes, 1999). But how can this mapping be established? We want to analyse this relation between perception and motor control through applying our simple body model in perception. As during this learning one has only access to the resulting

perceived visual input, the learning has to take place in an unsupervised manner. Therefore, the acquisition of such a mapping from seeing someone moving around to ones own movement systems seems quite difficult if not intractable. The main idea in our approach is that the introduction of the body model into the processing chain of perception dramatically simplifies the acquisition of a mapping. Both processes share the underlying body model in our setting and in this way the dynamic development of both processes is constrained in the same way. We want to show that this is enough to come up with a mapping and how this simplifies finding the mapping.

In figure 5 it is shown how the two models are connected and how they are incorporated into their respective system. On the left side, an acting agent is shown. Here the internal MMC network model is used in the same way as explained in the preceding section. A target value is set as an input to the model. The network is approaching a solution and at the same time moves the connected arm. The movement of the arm is perceived by the observing agent on the right sight. In a preprocessing step characteristic visual features are extracted from the visual image. The aim is to correlate the visual features with assumed body configuration. This has to be done in an unsupervised fashion as only the evolving visual features are available and the observer has no information on the segment vectors (only in the initial situation in which a resting position is assumed). But the observing agent can exploit its knowledge about the dynamics of the unknown segment vectors as this dynamics are shared between both agents and are encoded in the body model. The general idea is that the observing agent tries to hook the body model up to the visual features and close the loop in trying to predict the visual features. The underlying assumption is that the predicted change of the visual features can only be correctly produced by the dynamics of the observer's body

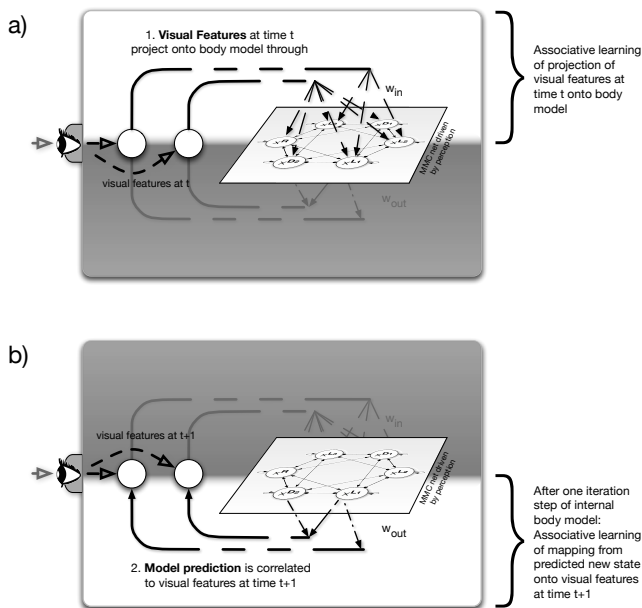


Figure 6: Steps when applying the internal body model in perception and how this allows for learning associations to visual features. At first (in a) visual features are associated towards the current body model activation. After one processing step of the body model connections back to perception are learnt which associate the new predicted values of the body model with the updated visual features. Initially the connections are random and only by accident correlations will occur. These will be strengthened over time and mappings between the two spaces evolve.

model when it is in a similar state as the actor's body model. We want to test this assumption for our simple model and in addition how easily this then allows to bootstrap the connection to visual features from the body model.

The internal model in the observer is used as a predictor. It can be regarded as a hidden mediating layer of a neural network linked to the visual features. The input layer are the visual features at a certain time  $t$  and the task would be to learn projections from these visual features to the body model (fig. 6 a). After one time-step in the mediating body model layer the activations of the body model should be routed back to the visual features which have new values now for time  $t + 1$ . This mapping should also be learnt (fig. 6 b) and as we only have access to the visual features—given as input and output—both mappings have to be learnt at the same time. The basic idea is that this is possible and that the correlation of the sequence of observed features is correlated with the body model dynamics. Hebbian-type learning should be sufficient to identify the associations and establish the mapping (Hebb, 1949). Figure 7 shows a different perspective on the whole network. The network is spread out

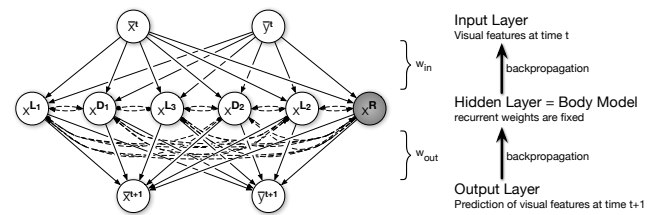


Figure 7: Schematic sketch of the network architecture used for learning the input and output mapping between visual features and body model. The recurrent connections of the hidden layer are fixed and setup as a MMC network. As the network shall be used in the same mode as when used for motor control the target vector  $R$  (right) corresponds to the predicted target position estimated from the known dynamics of the network.

into a three layer neural network. The input layer is given through the visual features at a certain point in time  $t$ . The body model constitutes the middle layer. As the dynamics of the two models are essential for establishing a coupling, this network must be driven in the same way as the original network. Therefore, the  $R$  vector (shaded in the figure) does not represent the current end effector position, but the target position of a movement. This is unknown to the network and the network can only observe the current state, but incorporating knowledge about the known dynamics the end state can be easily estimated (see (Schilling, 2009)). The weights are predetermined for the hidden layer as it represents the MMC network, but the activation of this layer is hidden during learning. The output layer represents the predicted output for one timestep later ( $t + 1$ ). Here we have simplified the view on the overall architecture as we are introducing this output level for representing the visual features at time  $t + 1$ . The back projection on the visual features in the overall framework is more complex as the function of these connections depend on the context. In perception the body model is not supposed to re-activate the visual features in general. But in specific cases it would be an advantage to use the prediction, e.g. when part of the movement can not be observed (the arm might move behind an object and is occluded for a short time). Therefore, it must be possible to use this connections in different ways depending on the context and inhibit their reactivation during perception. During learning these connections are essential for correlating the predicted state of the body model to the new visual features. Introducing these new visual features as separate units in an output layer allows us to come up with the simple general structure shown in fig. 7 and to use standard backpropagation-through-time learning (Rumelhart et al., 1986) to learn the two weight matrices at the same time. The feature values from one time step ahead can in this way be used as the target output values.

In a preprocessing stage visual features are extracted from

the perceived image of the arm. We use visual image moments. Image moments (Mukundan and Ramakrishnan, 1998) reflect characteristics of a foreground object in a given image. They capture the statistical regularities of the object pixel and describe in this way shape properties of the foreground object, e.g. size, orientation. The main advantages of image moments are that they provide a descriptive representation and at the same time are inexpensive to compute. They can be easily calculated from a binary pixel-based image with the intensity function  $I(x, y)$  where all object pixels are represented as a one and all other pixel have a value of zero:

$$M_{pq} = \sum_x \sum_y x^p y^q I(x, y) \quad (3)$$

Usually a set of image moments of different orders is used with the order of an image moment given as the sum of the two exponents  $p$  and  $q$  used in the equation above. The zeroth order moment is a count of the object pixel and from the first order image moments one can derive the visual center of gravity (COG, the centroid  $\bar{x}, \bar{y}$  of the object):

$$\bar{x} = \frac{M_{10}}{M_{00}}, \bar{y} = \frac{M_{01}}{M_{00}} \quad (4)$$

Higher order moments allow to compute orientation and shape properties of the object shown in the image.

We are only using the centroid information in our simulations. Using higher order image moments would of course allow for a better reconstruction of the visual image. But the focus of our work is on how the body model contributes to recognizing and tracking the seen arm. Relying only on insufficient information emphasizes the contribution of the body model.

The centroid information can be directly calculated from the segment vectors of the moving arm. The overall center of gravity is constituted as the mean of the individual segment visual COG (we assume uniform length and width of segments). The equations describing the segment COGs can be integrated through calculating the mean value:

$$\bar{x}^{ges} = \frac{1}{3}(\bar{x}^{L1} + \bar{x}^{L2} + \bar{x}^{L3}) \quad (5)$$

## Results

We want to mainly focus on the qualitative result that the network is able to establish input and output connections in a way that both networks activities are coupled. We used a simple back-propagation learning rule on a set of initially random weights. Back-propagation is known for depending on the initial configuration and converging onto local minima, therefore we started a series of simulations for different initial weights covering the whole space of weights. While in many simulations the network converged, it was not sufficiently able to predict the next visual features at all. The

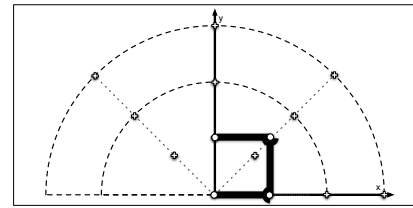


Figure 8: Shown is the initial configuration of the three segmented arm as solid black lines. The 12 targets are shown as white crosses.

network got stuck in a local minima. In these cases a constant value was returned or simply the input value. In the following we want to concentrate on the other simulations which were able to successfully predict the next visual features and want to look what the internal model was doing during predicting sequences. In general, the behaviour of all these networks was similar and in the following we use one example simulation series.

The network was trained from an initial arm configuration (shown in fig. 8 with all 12 targets). Both, the moving arm and the perceiving arm were initialised in this configuration, this means we assume for the simulations that there is a certain resting posture from which all movements start. There are 12 targets around this resting posture and we selected 9 for training and 3 later for testing on generalisation. The input and the output network (fig. 7) were then set to initial values. The perception network was trained on the visual data which resulted from the movement controlled by the movement network (see equation 5): a target was given to the movement network and the visual data before doing one iteration step in the movement control network was used as an input to the perception network. The visual features of the arm after the iteration step of the control network had been carried out was then used as the target value for the perception network which should learn to predict this value from the visual input. A movement lasted 15 iteration steps and the network was trained on a random order of the 9 training targets for 250 epochs (as mentioned above, the weights are not completely random, as we only cover a subset of the whole weight space here, see also discussion). The arm did not reach the target during the 15 iteration steps, but as the movement of the classical MMC is slowing down at the end, we only used the part of the movement containing rich dynamics, i.e. the arm is still considerably moving.

In fig. 9 a movement to an example target is shown for both arms. The body model used in the perception loop is following the leading moving arm. Both networks are in good agreement and synchronized. One advantage of the presented approach is that it extends also to movements not shown before as underlying knowledge about kinematics and body constraints is incorporated in the perception process. Figure 10 shows the behavior of the perception model

Movement of the arm towards (0, 0.3)

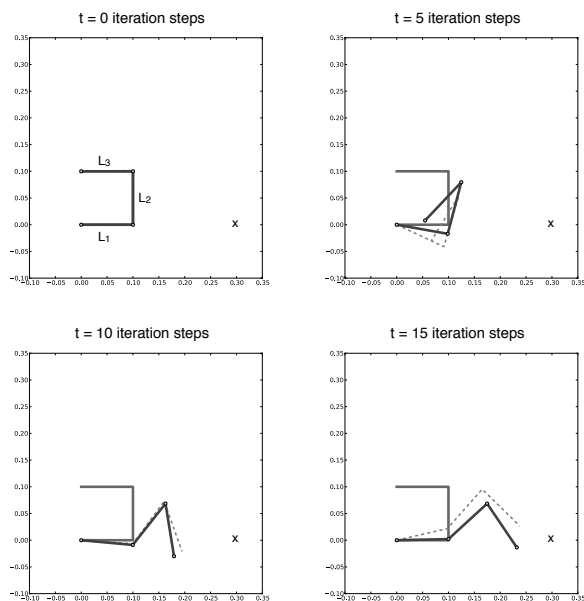


Figure 9: An example of the perceived arm movement. Course of time is going from left to right, top to down. Shown are snapshots of iteration 0, 5, 10 and 15. In the first figure at the left, top the initial configuration is shown in light gray. The moving arm is shown as a dashed line and the current state of the MMC model used for perception is represented as the dark grey line.

when the moving arm is approaching a novel target.

There was no observable difference between targets used for training and novel targets. During each test run the moving arm reached out during a period of 15 iteration steps towards one target from the initial configuration. We are not interested in finally reaching the target as during the last part of the movement the arm is only moving slowly for the classic MMC approach and the interesting part for our comparison is the comparison for the more dynamic starting phase. In general, the two networks converged for the final part of the movement to their respective endpoints. The observing network adopted in all cases a qualitatively similar configuration (as shown in the example, i.e. the segments of both networks are orientated in a similar way). We compared the differences of the single segment orientations to evaluate differences in configurations of the networks states. The difference angle for the segment orientations of the perceived arm and the moving arm were computed for each segment. The mean difference overall segments was 0.125 rad (standard deviation  $\pm 0.396$  rad). Mostly differences of the last segments were responsible for the high variation. This can be explained by the fact that the orientations of the first segment is weighted very high in the computation of the visual features.

## Discussion

We have shown first results indicating that sharing a common principle organizing movement dynamics is sufficient to bootstrap associations from the internal control network to visual features. After successful learning, the body model is coordinated with the motor control network solely through the simple visual features which in themselves would not be sufficient to estimate the manipulator configuration. Until now the simulation results are a first step providing a qualitative finding and the high variation is also a result of the simple visual features used to describe the postures.

One problem with the presented approach is that the simple back-propagation learning method on its own is not able to converge as the method depends on the initial configuration. Therefore, we started a series of simulations with different initial weight configurations (only for the input weights) covering large parts of the weight matrix space. To test that—in the successful cases—the success was not already predetermined through the selection of a suitable weight matrix, we tested the impact of the input network. Even in a supervised case it was not possible to learn the projections of the visual features onto the manipulator variables. The visual features in themselves do not carry enough information to predict the manipulator state. Therefore, the success of the network seems not given through the input transformation, but depends on the interplay between all the parts. In the future, we want to extend our approach and apply a more powerful learning algorithm (like a least-square method) which is able to overcome local minima and does not depend on the initial weight configuration. In addition, we want to perform a correlational analysis of the resulting weight matrices.

Other approaches to learn internal models of the body usually apply a supervised learning method. A nice example is the learning of a visual body model by Spranger (Steels and Spranger, 2008) in which a robot performs actions in front of a mirror and starts learning to associate the proprioceptive features to the observed visual features. Hoffmann et al. (2010) gives a thorough review about other approaches along the same line and on the integration of other modalities into the body schema in robots.

In the future, we will apply our approach in a real world robot scenario in which a robot is at first learning to recognize what another robot is doing when both apply their internal MMC-type body model. The task shall be implemented as a communicative scenario and in a second step a group of robots shall come up with a shared conceptualization of body postures through performing short interactions (language games (Steels and Belpaeme, 2005)). Besides additional preprocessing steps this would require to incorporate more descriptive visual features as are higher level centralized visual moments. The implementation on the robots is done in cooperation with the CSL group (Luc Steels, Paris). In the final system, each agent would have de-

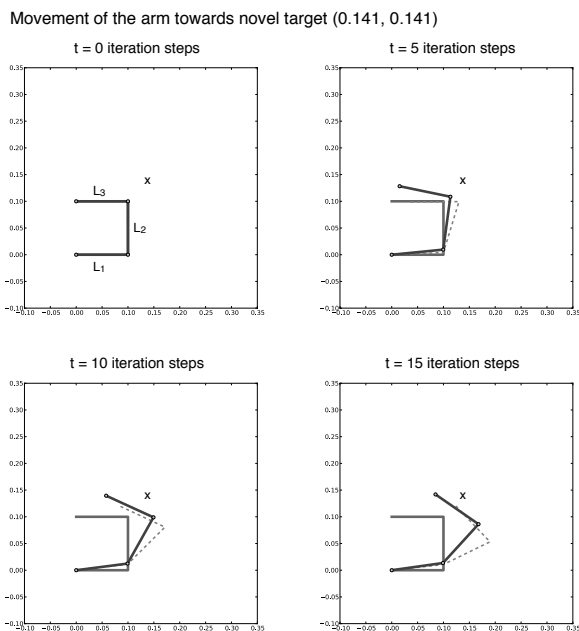


Figure 10: Movement to a novel target. Course of time is going from left to right in two rows. Shown are snapshots of iteration 0, 5, 10 and 15. In the first figure at the left, top the initial configuration is shown in light gray. The moving arm is shown as a dashed line and the current state of the MMC model used for perception is represented as the dark grey line.

veloped a conceptual space from a simple grounded internal representation of the own body which is now multimodal in its nature. Therefore, this internal model and the mapping onto visual features allow to be utilized in perceiving others making movements and coming up with conventional—and in a population agreed on—symbols. This would open the door for a simple form of communication and cooperation inside the rules given through the language game.

## Acknowledgements

This work was supported by a DAAD grant to Malte Schilling.

## References

- Bernstein, N. A. (1967). *The Co-ordination and regulation of movements*. Pergamon Press Ltd., Oxford.
- Cruse, H. and Steinkühler, U. (1993). Solution of the direct and inverse kinematic problems by a common algorithm based on the mean of multiple computations. *Biological Cybernetics*, 69:345–351.
- Decety, J. and Grezes, J. (1999). Neural mechanisms subserving the perception of human actions. *Trends in Cognitive Sciences*, 3(5):172–178.

- Grush, R. (2004). The emulation theory of representation: Motor control, imagery, and perception. *Behavioral and Brain Sciences*, 27:377–442.
- Hebb, D. O. (1949). *The Organization of Behavior*. John Wiley, New York.
- Hesslow, G. (2002). Conscious thought as simulation of behaviour and perception. *Trends in Cognitive Sciences*, 6(6):242–247.
- Hoffmann, M., Marques, H., Arieta, A. H., Sumioka, H., Lungarella, M., and Pfeifer, R. (2010). Body schema in robotics: a review. *IEEE Trans. Auton. Mental Develop.*, 2(4):304–324.
- Jeannerod, M. (2006). *Motor Cognition — What Action tells the Self*. Oxford: University Press.
- Makarov, V., Song, Y., Velarde, M., Hübner, D., and Cruse, H. (2008). Elements for a general memory structure: properties of recurrent neural networks used to form situation models. *Biological Cybernetics*, 98(5):371–395.
- Morasso, P. (1981). Spatial control of arm movements. *Experimental Brain Research*, 42(2):223–227.
- Mukundan, R. and Ramakrishnan, K. (1998). *Moment Functions in Image Analysis: Theory and Applications*. World Scientific, London, UK.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). *Learning internal representations by error propagation*, pages 318–362. MIT Press, Cambridge, MA, USA.
- Schacter, D. L., Addis, D. R., and Buckner, R. (2007). Remembering the past to imagine the future: the prospective brain. *Nature Reviews Neuroscience*, 8(7):657–661.
- Schilling, M. (2009). Dynamic equations in MMC networks: Construction of a dynamic body model. In *Proc. of The 12th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR)*.
- Schilling, M. (2011). Universally manipulable body models — dual quaternion representations in layered and dynamic MMCs. *Autonomous Robots*, 30(4):399–425.
- Schilling, M. and Cruse, H. (2007). Hierarchical MMC Networks as a manipulable body model. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2007)*, Orlando, FL, pages 2141–2146.
- Steels, L. (2003). Intelligence with representation. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 361(1811):2381–2395.
- Steels, L. and Belpaeme, T. (2005). Coordinating perceptually grounded categories through language: A case study for colour. *Behavioral and Brain Sciences*, 28(04):469–489.
- Steels, L. and Spranger, M. (2008). The robot in the mirror. *Connection Science*, 20(4):337–358.
- Steinkühler, U. and Cruse, H. (1998). A holistic model for an internal representation to control the movement of a manipulator with redundant degrees of freedom. *Biological Cybernetics*, 79(6):457–466.



# An animat's cell doctrine

Lisa Schramm<sup>1</sup> and Bernhard Sendhoff<sup>2</sup>

<sup>1</sup>Technische Universität Darmstadt, Karolinenplatz 5, 64289 Darmstadt, Germany  
<sup>2</sup>Honda Research Institute Europe, Carl-Legien-Str. 30, 63073 Offenbach, Germany  
lschramm@rtr.tu-darmstadt.de

## Abstract

We present a developmental model to simulate swimming digital organisms following an animat's cell doctrine. Morphology and control are encoded in one genome concurrently using artificial cells as the basic building blocks for both. Each individual starts with one cell in the middle of a computational environment, and its development is controlled by a gene regulatory network. The cells can differentiate into central pattern generators that control the movements of the resulting individual. After the developmental process, the individual is placed into a physics simulation environment and the distance it swims in a defined time is evaluated. Contrary to most existing models, one genome for both, morphology and control is used and the CPGs representing the dynamic control contribute to the morphology of the organism.

## Introduction

Following the work of Matthias Jakob Schleiden on plant tissues, Theodor Schwann postulated in 1839 that the tissue of all living organisms is made up of individual cells. At first this excluded the nervous system, which was later rectified by the seminal neuro-anatomical work of Ramón y Cajal and others. This principal concept is known as the cell or the neuron doctrine of biology.

In biology, the cell doctrine (including the nervous system) is an integral part of the evolution, development and operation of all living organisms. The cell as the carrier of the hereditary information is not just the basic functional unit of organisms, it is also the basic unit for the evolutionary process. Turning this argument around, we can hypothesize that the direction of the evolutionary process and its diverse results are a consequence of the cell doctrine. More strongly, evolution would have not been successful<sup>1</sup> without the cell as its basic unit. We also note that most of the evolutionary history has been devoted to single cell organisms rather than to multicellular ones.

<sup>1</sup>What does it mean, evolution being successful? To circumvent a philosophical discussion, we will resort to an artificial life perspective, equating success with progress in the criterion chosen for the process.

In artificial life, biological paradigms are frequently sought to facilitate the development of digital organisms or animats. The purpose of this paper is to outline a model that allows the simulation of digital organisms based on basic cell-like units, thus paving the way to an animat's cell doctrine including the nervous system or in more abstract terms the control system of the animat.

Since the seminal work of Karl Sims (Sims, 1994) the co-evolution of the morphology (=body) and the control systems (=brain) of digital organisms has received continuous attention. In Sim's work a developmental model using a directed graph has been used for both neural controller and body plan. The role of the morphology to reach a certain functionality has also been discussed in robotics. The passive walker (McGeer, 1990) demonstrated convincingly how the specific mechanical configuration alone can lead to a walking behavior that closely resembles the one we observe in humans without complex control algorithms. However, not least due to the mechanical difficulties the body is mainly unchanged in most evolutionary or developmental robotics approaches. Evolving the developmental steps of a controller in a static morphology has no justification and its limitations have been recognized, see e.g. (Pfeifer et al., 2007). Although some advances have been made using mechanical cell blocks to enable a changing morphology, the mechanical restrictions are still fundamental (Murata and Kurokawa, 2007; Meng et al., 2011).

In the digital world, we face much fewer restrictions and it is possible to simulate completely cell based animats, see e.g. (Schramm et al., 2009). Several computer models for brain-body co-evolution have been proposed in the literature, see e.g. (Hornby and Pollack, 2001; Miconi and Channon, 2006; Spector et al., 2007). However, models have either been detailed with regard to neural development (Kitano, 1995) or with the development of the morphology (Andersen et al., 2009; Eggenberger Hotz et al., 2003). Using a more abstract representation for the body morphology, Jones et al. (2011, 2008) analyzed the effects of the body plan on neural organization using energy constraints. Bongard and Paul (2000) studied the correlation between morphological

symmetry and locomotive efficiency using a direct encoding. The advantage of being able to evolve a bilaterally symmetric body plan or neural controller has been reported independently in (Mazzapioda et al., 2009; Oros et al., 2009). Bongard (2003) uses a gene regulatory model to develop locomoting animats or animats that should grow to touch an object.

A number of computational models have been developed to model biological gene regulatory networks (see e.g. the review of de Jong (2002)). Artificial embryogeny simulates biological cellular growth and pattern formation starting with one single cell (Andersen et al., 2009; Eggenberger Hotz et al., 2003; Harding and Banzhaf, 2008; Joachimczak and Wröbel, 2009; Doursat, 2009; Kowaliw et al., 2004). Steiner et al. (2008) evolved the structure and the parameters of a gene regulatory network for growing 3D cellular structures that are mechanically stable and lightweight. The model was refined in (Steiner et al., 2009) using cell polarization to represent more complex inner structures. Stanley and Miikkulainen (2003) develop a taxonomy for artificial embryogeny based on cell fate, targeting, heterochrony, canalization, and complexification.

In this contribution, we implement an animat’s cell doctrine by representing the whole body or morphology of the digital organism by cells some of which perform the control of the animat’s behavior. Therefore, the nervous system is an integral part of the morphology and the neurons are basic cells that differentiate during embryogeny assuming their specific neural functionality. Therefore, the system evolves the shape and the control of animats concurrently. Furthermore, the representations of shape and control are not separated, instead morphology and control are phenotypic characteristics of the artificial organisms that are the result of a common gene regulatory network that organizes the cellular growth of the animat. Indeed the separation between morphology and control becomes arbitrary even on the phenotypic level, because the cells that control the behavior also contribute to the morphology of the animat. This straightforwardly results from using artificial cells as the basic structural as well as functional components of our animat.

For the simulation of the cellular neural control we use central pattern generators (CPGs) which represent a higher level of abstraction compared to the spiking neural system employed in (Jin et al., 2008). CPGs facilitate the evolution of an oscillating movement, which makes it easier for the evolutionary process to develop the swimming behavior.

In the next section, we introduce CPGs in general and the specific CPG model used in this paper in greater detail. The following section is devoted to a description of our model of gene regulatory networks (GRNs) and how it is used to represent cellular growth. Thereafter, the physics simulation and the experiments are described followed by a discussion of the results. In the last section, the main findings of the paper are summarized and an outlook into future experiments

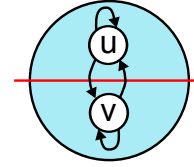


Figure 1: The model of a central pattern generator contains two neurons that interact with each other.

Table 1: Properties of the CPG Model

$k$	$\omega$	$\rho$	$\lambda$	$\sigma$
0.01	0.3	1	1	1

is presented.

## Central Pattern Generators (CPGs)

Many animals use coupled rhythmic muscle activations for movements. This movement is not controlled by the brain, but by coupled oscillators, the central pattern generators (CPG). It can be shown, that the pattern occurs also after the spinal cord has been separated from the brain (Murray, 2008).

Several models of CPGs exist, e.g. (Murray, 2008; Ijspeert and Kodjabachian, 1999; Verdaasdonk et al., 2006; Chung and Slotine, 2010; Beer, 2009), in general the CPG consists of two neurons which interact with each other, see Figure 1. The difficulty with most models is the stability of the output of many CPGs depending on their connections. The output of the CPGs should ideally be sinusoidal with phase shifts between the output signals of the different CPGs depending on their synapse connections and weights. Each CPG oscillates, they synchronize with other CPGs using their connections, so no global clock is used.

Chung and Slotine (2010) use coupled Hopf-Kuramoto oscillators and show their ability to synchronize almost globally. This model is used for the experiments presented in the following because of its good ability to synchronize. Therefore,  $\mathbf{x}_i(t) = (u_i(t), v_i(t))^T$  and the following equations are used:

$$\dot{\mathbf{x}}_i = \mathbf{f}(\mathbf{x}_i; \rho_i) - k \sum_{j \in \mathcal{N}_i}^{m_i} \left( \mathbf{x}_i - \frac{\rho_i}{\rho_j} \mathbf{R}(\phi_{ij}) \mathbf{x}_j \right) \quad (1)$$

and

$$\mathbf{f}(\mathbf{x}; \rho) = \begin{pmatrix} -\lambda/\rho^2 (u^2 + v^2 - \rho^2 \sigma) u - \omega(t)v \\ \omega(t)u - \lambda/\rho^2 (u^2 + v^2 - \rho^2 \sigma) v \end{pmatrix}. \quad (2)$$

The properties of the model for the simulations in this paper are described in Table 1.

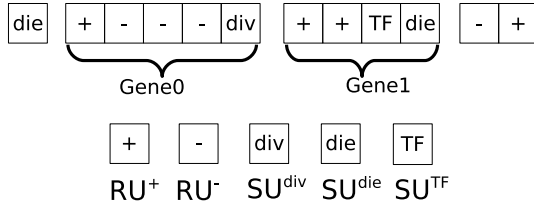


Figure 2: An example chromosome for the development. The first gene (gene 0) starts at the first RU of the genome. Each SU-RU changeover defines a boundary between two genes.

## A Computational Model for the Development of Morphology and Control

The morphological development simulated in this work is under the control of a gene regulatory network (GRN) and physical cellular interactions. The morphological development starts with a single cell put in the center of a two-dimensional computational area of size  $100 \times 80$ . Each cell can die or divide. The cells are not fixed on a grid and underlie physical interactions, i.e. overlapping cells push each other away and cells that do not overlap attract each other with decreasing forces with larger distances.

The GRN is defined by a set of genes, each consisting of a number of regulatory units (RUs) and structural units (SUs). SUs define cellular behaviors, such as cell division, cell death or the production of transcription factors (TFs) for intra- and inter-cellular interactions. Whether the SUs of a gene are expressed is determined by the activity level of the RUs of the gene, refer to Fig. 2. Note that a single or multiple RUs may regulate the expression of a single or multiple SUs and that RUs can be activating ( $RU^+$ ) or repressive ( $RU^-$ ). The activation level of RUs is influenced by the TFs that can “bind” to the RU. If the difference between the affinity values of a TF and a RU is smaller than a predefined threshold  $\epsilon$  (in this work  $\epsilon$  is set to 0.2), the TF can bind to the RU to regulate the gene activation. The affinity values are encoded in the RUs and the SUs that produce a TF and are, as well as all values in the genome, limited to an interval of  $[0, 1]$ . The affinity similarity ( $\gamma_{ij}$ ) between the  $i$ -th TF and  $j$ -th RU is defined by:

$$\gamma_{i,j} = \max(\epsilon - |\text{aff}_i^{\text{TF}} - \text{aff}_j^{\text{RU}}|, 0). \quad (3)$$

If  $\gamma_{i,j}$  is greater than zero, then the concentration  $c_i$  of the  $i$ -th TF is checked whether it is above a threshold  $\vartheta_j$  defined in the  $j$ -th RU:

$$b_{i,j} = \begin{cases} \max(c_i - \vartheta_j, 0) & \text{if } \gamma_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases}. \quad (4)$$

Thus, the activation level contributed by the  $j$ -th RU (de-

noted by  $a_j, j = 1, \dots, N$ ) can be calculated as follows:

$$a_j = \sum_{i=1}^M b_{i,j}, \quad (5)$$

where  $M$  is the number of TFs that bind to the  $j$ -th RU. Assume the  $k$ -th gene is regulated by  $N$  RUs, the expression level of the gene can be defined by

$$\alpha = g(c), \quad (6)$$

$$g_k(c) = 100 \sum_{j=1}^N l_j a_j (2s_j - 1), \quad s_j \in (0, 1). \quad (7)$$

$2s_j - 1$  denotes the sign (positive for activating and negative for repressive) of the  $j$ -th RU and  $l_j$  is a parameter representing the strength of the  $j$ -th RU. If  $\alpha_k > 0$ , then the  $k$ -th gene is activated ( $\delta_k = 1$ ) and its corresponding behaviors coded in the SUs are performed.

An SU that produces a TF ( $SU^{\text{TF}}$ ) also encodes all parameters related to the TF, such as the affinity value, the decay rate  $D_i^c$ , the diffusion rate  $D_i^f$ , as well as the amount of the  $TF_i$  to be produced. Which  $TF_i$  is produced is defined in terms of the affinity value.

$$\begin{aligned} \mathbf{A} &= \mathbf{h}(\alpha), \\ h_i(\alpha_k) &= \begin{cases} \beta \left( \frac{2}{1 + e^{-20 \cdot f \cdot \alpha_k}} - 1 \right) & \text{if } \alpha_k > 0 \\ 0 & \text{otherwise} \end{cases}, \end{aligned} \quad (8)$$

where  $f$  and  $\beta$  are both encoded in the  $SU^{\text{TF}}$ .

A TF produced by an SU can be partly internal and partly external. To determine how much of a produced TF is external, a percentage ( $p^{\text{ext}} \in (0, 1)$ ) is also encoded in the corresponding gene. Thus,  $\Delta c_i^{\text{ext}} = p^{\text{ext}} \cdot A_i$  is the amount of external TF to be produced and  $\Delta c_i^{\text{int}} = (1 - p^{\text{ext}}) \cdot A_i$  is that of the internal TF.

External TFs are put on four grid points around the center of the cell, which undergo first a diffusion and then a decay process. Note, that the external TFs are computed on a grid but the positions of the cells are continuous and therefore not limited to this grid. The internal TFs underlie only a decay process. All internal and external concentrations of TFs are limited to an interval of  $[0, 1]$ .

Figure 3 shows a block diagram of the main components of a GRN in one cell, describing the cell dynamics. The cell dynamics can become coupled through external transcription factors, which underlie a diffusion and decay process and are position dependent. The number of TFs involved in gene regulation of the cellular behaviors is defined by the genome and the parameters in the resulting GRN as well. The number of cells also changes during development, starting with one single cell and two external TFs. The maximum number of cells is limited to 700 cells for reducing computational cost. From a control system point of view, the developmental system is composed of a changing number of

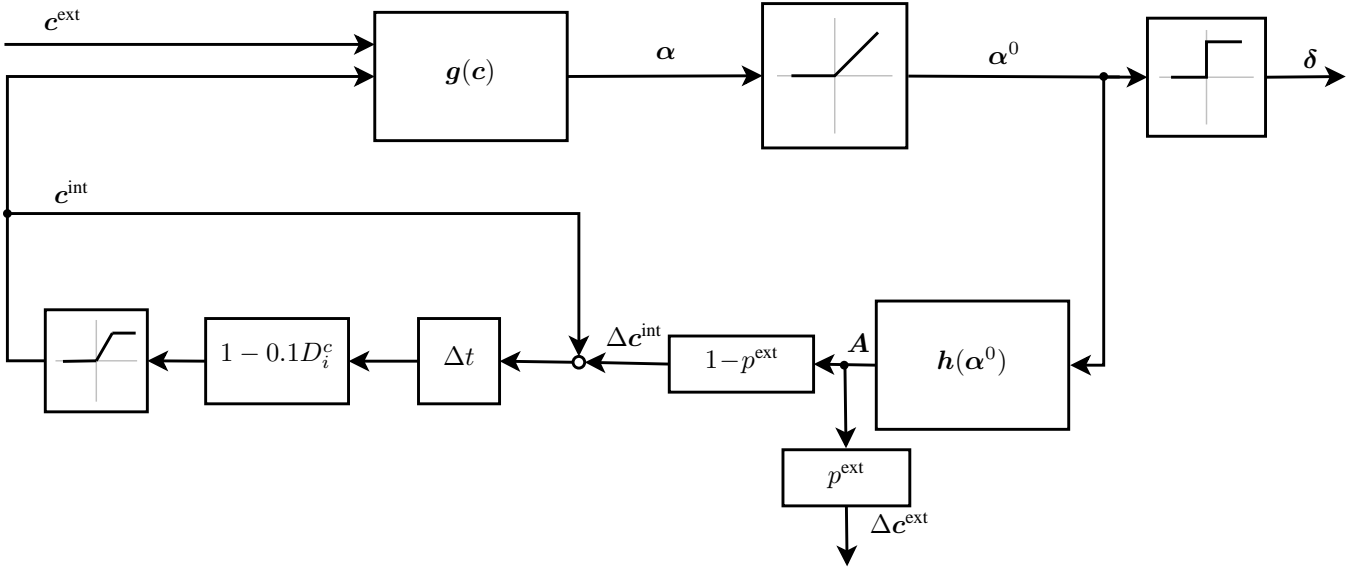


Figure 3: Block diagram of the model of a single cell.

nonlinear dynamical sub-systems with a changing number of system states, and the dynamics of the sub-systems are strongly coupled with each other.

In our experiments, we put two prediffused, external TFs without decay and diffusion in the computation area. The first TF has a constant gradient in the  $x$ -direction and the second in  $y$ -direction.

The SU for cell division ( $SU^{\text{div}}$ ) encodes the angle of division, indicating where the daughter cell is placed. A cell with an activated SU for cell death ( $SU^{\text{die}}$ ) dies at the developmental timestep it is activated. When both cell death and cell division are active at the same developmental step, only cell death is performed.

A cell with an active SU for neuron formation ( $SU^{\text{neuron}}$ ) becomes a CPG for the rest of its lifetime. All cells on the outside of the individual that are not CPGs at the end of the development are termed muscle cells. The threshold for whether the  $i$ -th CPG is to be connected to the  $j$ -th CPG is calculated as follows:

$$\varphi_{ij} = \frac{c_1}{1 + e^{c_2 \cdot (d_{ij} - 10c_3)}}, \quad (9)$$

where  $d_{ij}$  is the distance between the  $i$ -th and  $j$ -th neuron and  $c_1$ ,  $c_2$  and  $c_3$  are encoded in the  $SU^{\text{neuron}}$ . Then, a random number  $p$  ( $p \sim \mathcal{N}(0, 1)$ ) is generated, and if  $p < \varphi_{ij}$ , a connection between the two CPGs will be generated.

There is one additional SU for other possible actions, which are not used in this work. As a result, it can happen that some genes perform no action, that is one cause of redundancy.

The muscle cells contract with the output of one of the neurons of the closest CPG. When the distance to the closest CPG is higher than 8, the muscle cell is passive. A contrac-

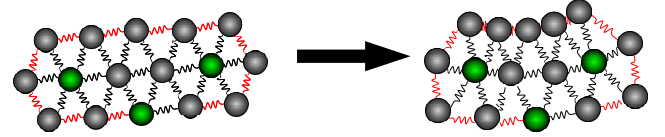


Figure 4: Illustration of a body plan consisting of cells connected by springs. The CPGs are depicted in green. The springs on the outside of the body (red) are able to change their natural length, except the springs associated to a CPG.

tion of a muscle cell means a change in the rest length of the associated spring at the outside of the individual (counter-clockwise).

Since each CPG contains two neurons ( $u$  and  $v$ ), an orientation of the CPG is introduced to define to which neuron a cell is connected. The orientation of the CPG itself is defined by the gradient of a TF, which TF is used is defined in the SU for neuron formation. Parameter  $s_4$  in the SU defines an affinity value, the TF with the closest affinity to the affinity encoded in  $s_4$  is used for the orientation of the CPG. Cells which connect to the CPG on its first  $0 - 180^\circ$  are connected to the neuron  $u$  and cells connected with an angle of  $180 - 360^\circ$  are connected to the neuron  $v$  of the CPG.

## Physics Simulation

The physics simulation engine used to simulate the behavior of the animats is BREVE <sup>2</sup>

A simple model for simulating the effects of water forces is added, which has also been adopted in (Sfakiotakis and Tsakiris, 2006). In this model, the water forces for different

<sup>2</sup>see [www.spiderland.org/](http://www.spiderland.org/)

Table 2: Constants for the mechanical simulation environment

Mass of cells $m$	0.5
Radius of cells $r$	0.5
Damping constant $d$	1
Spring strength $c$	5
Normal natural length of springs $l_n$	2
Short natural length of springs $l_s$	1.2
Minimal periodic time $T_{min}$	10
Maximal periodic time $T_{max}$	400
Simulation length $t_{sim}$	500.0

elements  $i$  (sphere of the  $i$ -th cell) are computed as follows:

$$\mathbf{F}^i = \mathbf{F}_T^i + \mathbf{F}_N^i, \quad (10)$$

$$\mathbf{F}_T^i = -\lambda_T \cdot \text{sgn}(\mathbf{v}_T^i) \cdot (\mathbf{v}_T^i)^2, \quad (11)$$

$$\mathbf{F}_N^i = -\lambda_N \cdot \text{sgn}(\mathbf{v}_N^i) \cdot (\mathbf{v}_N^i)^2, \quad (12)$$

where  $\lambda_T$  and  $\lambda_N$  are the drag coefficients for each direction.  $\lambda$  depends on the effective area, a shape coefficient of the element and the fluid density.  $\mathbf{v}_T^i$  and  $\mathbf{v}_N^i$  are the velocities of element  $i$  in normal and tangential direction.  $\lambda_T = 0.001$  and  $\lambda_N = 2.5$  are used in this work. The water forces are computed for cells in the outside of the body plan. The normal and tangential vectors of the body parts ( $i$ -th sphere) can be calculated by:

$$\mathbf{t}^i = \frac{\mathbf{p}^{i-1} - \mathbf{p}^{i+1}}{|\mathbf{p}^{i-1} - \mathbf{p}^{i+1}|}, \quad (13)$$

$$\mathbf{n}^i = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \cdot \mathbf{t}^i, \quad (14)$$

where  $\mathbf{p}^i$  is the position vector of the  $i$ -th cell and  $\mathbf{p}^{i-1}$  and  $\mathbf{p}^{i+1}$  are the positions of the neighboring cells on the outside of the morphology.

$$\mathbf{v}_N^i = \mathbf{n}^i \cdot \mathbf{v}^i, \quad (15)$$

$$\mathbf{v}_T^i = \mathbf{t}^i \cdot \mathbf{v}^i, \quad (16)$$

where  $\mathbf{v}^i$  is the velocity of the  $i$ -th cell.

## Experiments

The goal of the experiments is to evolve individuals that swim the furthest in a desired time. The fitness function for swimming is defined as follows:

$$f_{swim} = - \left| \left( \sum_{i=0}^n \mathbf{x}^i(t=0) \right) - \left( \sum_{i=0}^n \mathbf{x}^i(t_{end}) \right) \right|, \quad (17)$$

so the center of mass of the individual at the beginning and the end of the swimming period are computed and the distance is calculated.

Table 3: Properties of the evolutionary optimization

$\mu$	45
$\lambda$	300
Elitists	3
initial # RUs and SUs	50, 50
$\sigma$	$10^{-4}$
$p_{dup}, p_{trans}, p_{del}$	0.05, 0.03, 0.02

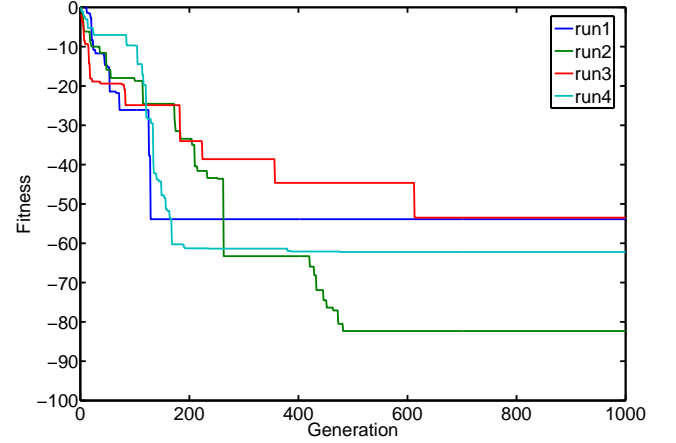


Figure 5: Fitness curves to evolve swimming individuals, their movements are controlled by CPGs.

The size of the individuals is limited, so the number of cells ( $n_c$ ) is constrained between 10 and 500. A penalty of  $600 - n_c$  will be applied if  $n_c < 10$  and a penalty of  $n_c$  if  $n_c > 500$ . If the cells in the developed morphology are not fully connected, a poor fitness of 100 will be assigned.

When the individual consists only of neurons or has no neurons, there will be no movement and the fitness for swimming is therefore set to zero ( $fit_{swim} = 0$ ). If the CPGs are not connected, which means there is no path to another CPG via synapses, the CPGs cannot synchronize and their phase shift is random and therefore depends on the initial values of the differential equation. To avoid that not connected CPGs get established during the evolution, but still not to penalize it too strong, the fitness for swimming is then halved.

The EA setup is defined in Table 3, four different runs with different random seeds have been performed.

## Results

The fitness curves of the four different runs are shown in Figure 5. The resulting individuals all swim between 53 and 82 length units (53.9, 82.3, 53.5, 62.2). Run 2 is analyzed in more detail in the following section.

### Analysis of Run 2

The fitness curve and the morphologies of some individuals from run 2 are shown in Figure 6. An elongated shape de-

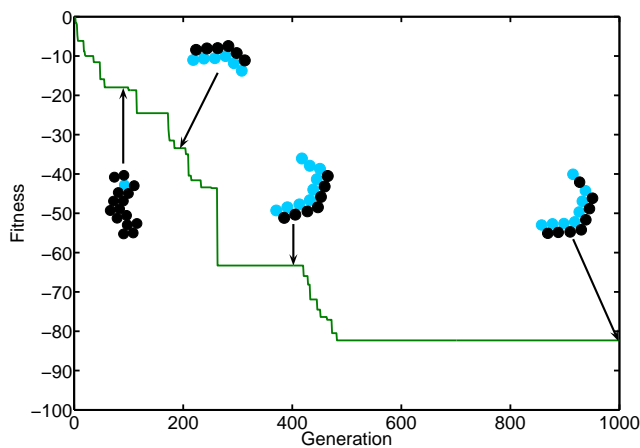


Figure 6: Fitness curve of run 2. The morphologies of the best individuals of generation 90, 200, 400 and 999 are drawn.

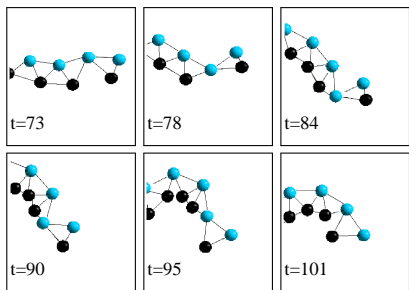
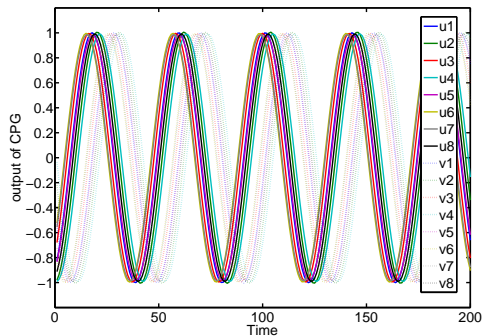


Figure 9: Tail fin of the best individual of run 2. Blue cells are CPGs, all other cells are black.

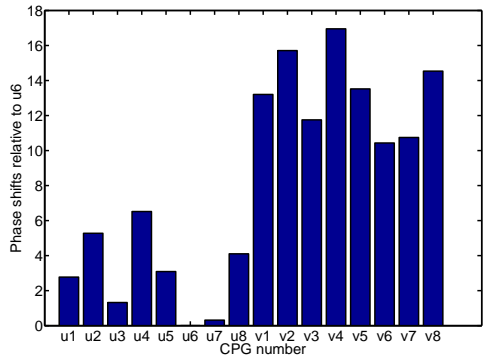
velops quickly (generation 90), and subsequently the shape smoothens in later generations. The number of the CPGs also increases and their positions change.

Figure 7 shows the development of the best individual of run 2, while Figure 8 shows its swimming behavior. Most cells first divide, transform to a CPG and die afterwards. Because of the neurons on one side of the individual, the springs on this side do not change their natural length and the movement of the individual is only caused by the springs on the other side of the individual. At the end of the individual a triangle forms which has the appearance and seems to fulfill the function of a tail fin, as shown in Figure 9. It is also interesting that the resulting individual is unsymmetric, contrary to the results of Jones et al. (2008) that show the advantage of symmetric morphologies.

The output of the CPGs are plotted in Figure 10, which shows that the phase shifts between the different CPGs are small. Figure 11 shows the orientations of the CPGs and we can see that some CPGs are turned around which results in a larger phase shift for the muscle cells.



(a) Time series of the different neurons.



(b) Phase shifts between the sinus curves of the different neurons relative to u6.

Figure 10: Output of all CPGs from the best individual of run 2.

## Summary and Outlook

In this paper, we have proposed a model that follows an animat's cell doctrine, i.e., an evolved gene regulatory network controls the cellular growth of a digital organism whose behavioral control is realized by some of the cells differentiating into central pattern generator cells representing neurons. Therefore, morphology and control of the animat are not merely co-evolved but co-represented by one regulatory system whose parameters are optimized during the evolutionary search process. Both on the genotypic and on the phenotypic level the distinction between morphology and control merely becomes descriptive.

The evolutionary optimization of the gene regulatory network resulted in a simple animat that is capable to perform swimming behavior by plausible movement. Body cells that differentiated into central pattern generators provide the ability to obtain an oscillating pattern with only a few neurons without limiting the connections or requiring long learning phases. In some cases the evolved morphology includes structures resembling tail fins, which seem to ease the functional or behavioral task. In principle, this is similar to the example of the passive walker that we mentioned in the introduction, where the dynamic control is eased by the

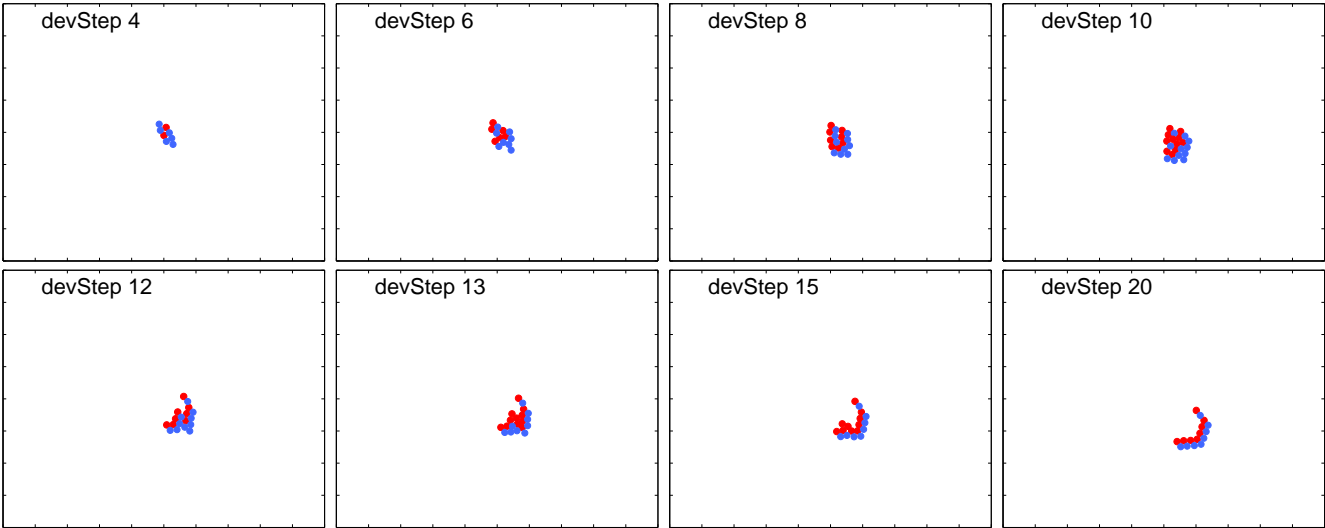


Figure 7: The development of the best individual of run 2 at the end of the evolution. Blue cells will divide in the next timestep, red cells transform to a CPG and will die in the next timestep.

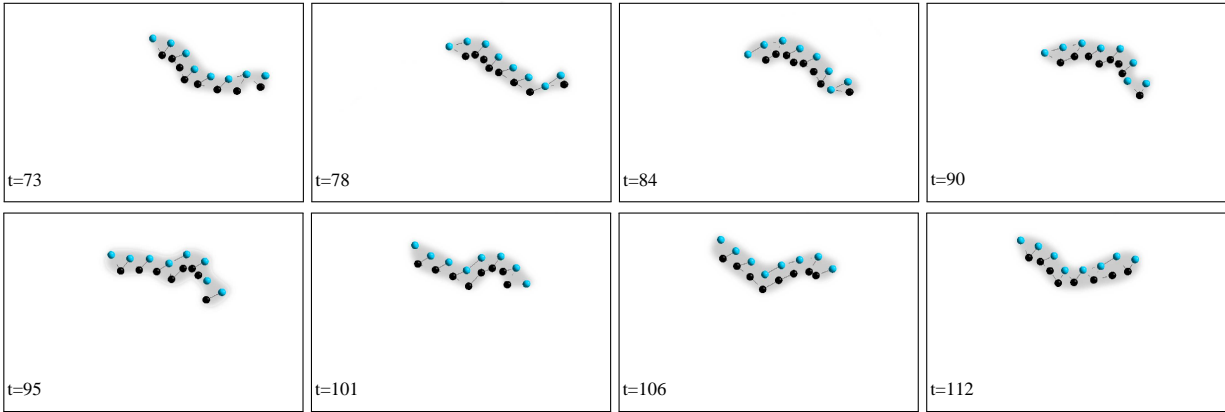


Figure 8: Swimming behavior of the best individual of run 2. Blue cells are CPGs, all other cells are black.

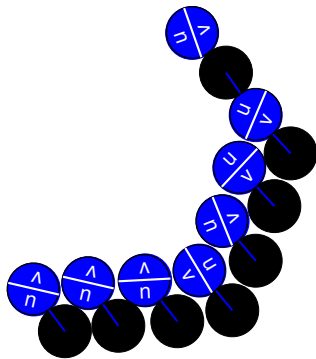


Figure 11: Orientation of the CPGs of the best individual of run 2. Since some CPGs are turned around, some muscle cells are connected to  $u$  and some to  $v$ , which causes the large phase shifts between the contractions of the springs.

morphology of the organism.

Compared to the work of Jones et al. (2011, 2008), which is based on a more abstract representation which is less biologically inspired, the evolved organisms do not exhibit symmetric morphologies. It would be interesting to find out under which constraints symmetry would also evolve in our framework.

The target of this research has been to demonstrate that the evolution of organisms exhibiting simple but meaningful behavior based on an animat's cell doctrine is possible. Finding the right parametrization of the gene regulatory network to develop a suitable morphology that incorporates the adjusted neural control is not a trivial task. At the same time, it is now necessary to analyze the properties of our model in more detail. First steps have been made in Figure 6 where we have observed the evolutionary path of the morphology for one run and in Figure 10 where we have analyzed how



the control is organized with the central pattern generator neurons. One of the next steps would be to relate the evolution of morphology to the evolution of the dynamics of the CPGs and how both are over time represented in the gene regulatory network. Unfortunately, even for digital evolution, the functional analysis of gene regulatory networks is a rather complex task, although promising first results have been obtained for evolving cellular morphologies, see e.g. Schramm et al. (2010).

## Acknowledgements

This work was supported by the Honda Research Institute Europe.

## References

- Andersen, T., Newman, R., and Otter, T. (2009). Shape homeostasis in virtual embryos. *Artificial Life*, 15(2):161–183.
- Beer, R. D. (2009). *Beyond Control: The Dynamics of Brain-Body-Environment Interaction in Motor Systems*, pages 7–24. Springer.
- Bongard, J. (2003). *Incremental Approaches to the Combined Evolution of a Robot's Body and Brain*. PhD thesis, University of Birmingham.
- Bongard, J. C. and Paul, C. (2000). Investigating morphological symmetry and locomotive efficiency using virtual embodied evolution. In *Proceedings of the SAB 2000*, pages 420–429.
- Chung, S.-J. and Slotine, J.-J. (2010). On synchronization of coupled hopf-kuramoto oscillators with phase delays. *arXiv:1004.5366v1*.
- de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103.
- Doursat, R. (2009). Facilitating evolutionary innovation by developmental modularity and variability. In *Proc. of the GECCO 09*, pages 683–690.
- Engenberger Hotz, P., Gómez, G., and Pfeifer, R. (2003). Evolving the morphology of a neural network for controlling a foveating retina - and its test on a real robot. In *Proc. of the ALife VIII*, pages 243–251.
- Harding, S. and Banzhaf, W. (2008). Artificial development. In Würz, R. W., editor, *Organic Computing*, chapter 9, pages 201–219. Springer.
- Hornby, G. and Pollack, J. (2001). Evolving L-systems to generate virtual creatures. *Computers & Graphics*, 25:1040–1048.
- Ijspeert, A. J. and Kodjabachian, J. (1999). Evolution and development of a central pattern generator for the swimming of a lamprey. *Artificial Life*, 5(3):247–269.
- Jin, Y., Schramm, L., and Sendhoff, B. (2008). A gene regulatory model for the development of primitive nervous systems. In *Proc. of the ICONIP 2008*, pages 48–55.
- Joachimczak, M. and Wröbel, B. (2009). Evolution of the morphology and patterning of artificial embryos: scaling the tricolour problem to the third dimension. In *Proc. of the ECAL 2009*.
- Jones, B., Jin, Y., Sendhoff, B., and Yao, X. (2008). Evolving functional symmetry in a three dimensional model of an elongated organism. In *Proc. of the ALife X*, pages 305–312.
- Jones, B., Soltoggio, A., Yao, X., and Sendhoff, B. (2011). Evolution of neural symmetry and its coupled alignment to body plan morphology. In *Proc. of the GECCO 2011*.
- Kitano, H. (1995). A simple model of neurogenesis and cell differentiation based on evolutionary large-scale chaos. *Artificial Life*, 2(1):79–99.
- Kowaliw, T., Grogono, P., and Kharma, N. (2004). Bluenome: A novel developmental model of artificial morphogenesis. In *Proceedings of GECCO 04*, pages 93–104.
- Mazzapioda, M., Cangelosi, A., and Nolfi, S. (2009). Co-evolving morphology and control: A distributed approach. In *Proc. of the CEC 2009*, pages 2217–2224.
- McGeer, T. (1990). Passive dynamic walking. *International Journal of Robotics Research*, 9(2):62–82.
- Meng, Y., Zhang, Y., Sampath, A., Jin, Y., and Sendhoff, B. (2011). Cross-ball: A new morphogenetic self-reconfigurable modular robot. In *Proc. of the ICRA 2011*.
- Miconi, T. and Channon, A. (2006). An improved system for artificial creatures evolution. In *Proceedings of Artificial Life X*, pages 255–261.
- Murata, S. and Kurokawa, H. (2007). Self-reconfigurable robots. *Robotics & Automation Magazine, IEEE*, 14:71–78.
- Murray, J. D. (2008). *Mathematical Biology I: An Introduction*. Springer, third edition.
- Oros, N., Steuber, V., Davey, N., Cañamero, L., and Adams, R. (2009). Evolution of bilateral symmetry in agents controlled by spiking neural networks. In *IEEE Symposium on Artificial Life*, pages 116–123.
- Pfeifer, R., Lungarella, M., and Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *Science*, 318(5853):1088–1093.
- Schramm, L., Jin, Y., and Sendhoff, B. (2009). Emerged coupling of motor control and morphological development in evolution of multi-cellular animats. In *Proc. of the ECAL 2009*.
- Schramm, L., Martins, V. V., Jin, Y., and Sendhoff, B. (2010). Analysis of gene regulatory network motifs in evolutionary development of multicellular organisms. In *Proc. of the ALife XII*, pages 133–140.
- Sfakiotakis, M. and Tsakiris, D. P. (2006). Simuun: A simulation environment for undulatory locomotion. *International Journal of Modelling and Simulation*, 26(4):350–358.
- Sims, K. (1994). Evolving virtual creatures. In *Proceedings SIGGRAPH*, pages 15–22.
- Spector, L., Klein, J., and Feinstein, M. (2007). Division blocks and the open-ended evolution of development, form, and behavior. In *Proceedings of GECCO*, pages 316–323.
- Stanley, K. O. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130.
- Steiner, T., Jin, Y., and Sendhoff, B. (2008). A cellular model for the evolutionary development of lightweight material with an inner structure. In *Proc. of the GECCO'08*, pages 851–858.
- Steiner, T., Trommler, J., Brenn, M., Jin, Y., and Sendhoff, B. (2009). Global shape with morphogen gradients and motile polarized cells. In *Proceedings of the 2009 Congress on Evolutionary Computation*, pages 2225–2232.
- Verdaasdonk, B. W., Koopman, B. F. J. M., and Helm, F. C. V. D. (2006). Energy efficient and robust rhythmic limb movement by central pattern generators. *Neural Networks*, 19(4):388–400.

# Feeding the Beast: Can Computational Demographic Models Free Us from the Tyranny of Data?

Eric Silverman<sup>1</sup>, Jakub Bijak<sup>1,2</sup> and Jason Noble<sup>3</sup>

<sup>1</sup>School of Social Sciences

<sup>2</sup>Centre for Population Change

<sup>3</sup>Institute for Complex Systems Simulation

University of Southampton

e.silverman@soton.ac.uk

## Abstract

Since its inception, ALife has moved from producing large numbers of highly-idealised, theoretical models towards greater integration with empirically collected data. In contrast, demography — the interdisciplinary study of human populations — has been largely following the principles of logical empiricism, with models driven mainly by data, and insufficient attention being paid to theoretical investigation. Such an approach reduces the ability to produce micro-level explanations of population processes, which would be coherent with the phenomena observed at the macro level, without having to rely on ever-increasing data demands of complex demographic models. In this paper we argue that by bringing ALife-inspired, agent-based methods into demographic research, we can both develop a greater understanding of the processes underlying demographic change, and avoid a limiting over-dependence on potentially immense sets of data.

– But you are paying a lot of money for the dragon!  
– And what, should we just give it to the citizens instead? [...] I see you know nothing about the principles of economics! Export credit warms up the economy and increases the global turnover.

– But it also increases the dragon as such – I stopped him. – The more intensely you feed him, the bigger he gets; and the bigger he gets, the higher his appetite. What kind of a calculation is it? He will finally devour you all!

Stanisław Lem, *Pożytek ze smoka* [*The Use of a Dragon*] (1983/2008: 186)

## Introduction

After attending the very first Artificial Life conference in 1987, the evolutionary biologist John Maynard Smith famously quipped that ALife appeared to be “fact-free science”. His comment was made in response to early ALife work (see, e.g., Langton, 1989) that tended to be abstract and conceptual, not to mention ontologically ambitious, making no connection to empirical data in the conventional sense.

Over time, the early enthusiasm for highly abstract models in ALife has lessened somewhat, as it has become increasingly clear that making such models empirically relevant involves a highly contentious theoretical commitment

to artificial life as an instantiation of biological life (Silverman and Bullock, 2004). Instead, abstract and conceptual ALife models have come to be viewed as tools for theoretical enquiry (Di Paolo et al., 2000), i.e., ways of explaining the qualitative dynamics of complex systems. At the same time, some modellers under the ALife banner have moved toward a greater connection with empirical data (e.g., Toquenaga et al., 1995; Smith V., 2008). ALife has experienced greater scientific respectability, we maintain, due to the collective recognition that modelling and simulation stand alongside theory generation and data collection in the scientific cycle — or, as Rossiter et al. (2010) put it, models are “first class citizens of science”.

Thus, ALife has been in a somewhat unique position: starting from methods almost completely disconnected from empiricism, the field has gradually moved toward a greater integration with empirical data, while retaining a focus on using simulation as a tool for theoretical investigation. In this paper we consider a discipline which appears to be following the opposite trajectory. Demography — the interdisciplinary study of the development of human populations — has long been a field devoted to predictive statistical modelling based on vast storehouses of data, while theory-building has mostly taken a back seat.

Demography’s intense devotion to data has served the field well when making projections of future demographic change in human populations. Nevertheless, traditional demographic methods struggle to develop well-founded *explanations* of these changes, going beyond simple generalisation of the observables (Burch, 2003). One of the motivations driving ALife’s shift toward greater connection to empirical data has been the recognition that neither theory nor data alone are enough to provide coherent explanations of phenomena. ALife has addressed this dilemma by bringing more data into a largely theory-focused modelling enterprise, and we propose that demography, in order to develop beyond its current epistemological limits, must also make a move towards the centre by incorporating conceptual and theoretical investigation into its heavily data-focused framework.

The scientific benefit to such an approach would be the enrichment of the theoretical foundations of demography. In this paper however we will also discuss another, perhaps more pragmatic, advantage to ALife-inspired demographic models: as a means for escaping some of the burdens of the time-consuming and combinatorially expensive data collection required to continue in the traditional fashion.

We begin our discussion in the next section with a summary of demography's struggles with its data-collection demands. We then move on to suggesting some potential applications of agent-based models for demographic research, describing the relevant strengths and weaknesses of the approach. Next, a detailed analysis of several demographic simulation models allows us to develop a more nuanced understanding of how agent-based models may provide new utility and insight. Finally, we offer our conclusions, and suggest some directions for future work in this area.

### Motivation: Meet The Beast

In the context of large, policy-focused projects in social science, modelling and simulation in some form has become ever more important as a means of providing useful information to stakeholders. Models provide a means of producing predictions or characterisations of complex systems which can give the stakeholder what they need: a target number, a summary of current numbers, or numbers to be wary of. However, many such modelling projects can become quite large and unwieldy. We often find that we require extensive amounts of data in order to feed into a large-scale model (hereafter, for illustrative purposes, referred to as 'the beast'), and the process of collecting that data is inevitably expensive and time-consuming. Plus, as our models get increasingly complex, the beast becomes ever hungrier.

Demography offers a unique predictive potential given the information embodied in the age structure of populations. However, for reasons we will discuss later, these predictions still remain largely uncertain. In an effort to alleviate some of the epistemological limitations, recent work in demography has attempted to bridge the gap between micro- and macro-level analysis (Courgeau 2007 and the MicMac project — see Willekens 2005 and Zinn et al. 2009). Advances in event-history analysis and microsimulations linked with multilevel statistical analysis have been offered as potential solutions to the micro-macro divide. However, such methods still have one major weakness: potentially enormous requirements for data due to the 'combinatorial explosion' of the parameter space. So, even as extended modelling frameworks, such as MicMac, try to bridge the micro-macro gap by producing linked simulations at both levels of analysis, we still find ourselves hamstrung by the need for large amounts of data.

Thus, we see demography reaching for more sophisticated modelling paradigms and for ways to produce more micro-level explanations of factors that drive population

change. Unfortunately, current modelling methods require us to continue 'feeding the beast': pumping models full of ever-increasing amounts of data, each dataset requiring vast amounts of resources (time and money) to collect. This has further impacts on the overall modelling enterprise: turn-around time for producing models grows out of control; stakeholders find themselves confronted with high-incomprehensible models and endless reams of data; and the primacy of post-hoc statistical analyses inevitably leads us toward certain types of models which seem to fit the data well.

In the other part of the methodological spectrum, the use of agent-based models à la ALife in recent years has become increasingly popular in certain areas within the social sciences. Starting from Schelling's (1978) famous residential segregation model, and moving on to Axelrod's *Complexity of Cooperation* (1984), Cederman's (1997) work on international relations, and the current wide spectrum of agent-based models in social science (cf. Epstein, 2008 or Gilbert and Troitzsch, 2005), the prospect of using agents to examine properties of human societies which are difficult or impossible to measure empirically has become increasingly attractive. Understandably, many social scientists are excited by the possibility of examining fundamental properties of social phenomena without being forced to devote excessive resources to primary data collection.

To date, much of the work extolling the virtues of agent-based models for the social sciences have focused on the potential explanatory benefits (Epstein, 2008). After all, by examining the processes occurring between agents, perhaps we can gain a greater understanding of how macro-level societal effects happen (although even this is debatable; see Sawyer 2005). However, we feel that another, perhaps more immediate benefit of agent-based modelling has been largely ignored in the literature: the prospect of escaping the expensive, time-consuming process of continual data collection.

Thus, in this paper we propose that agent-based models, informed by work in artificial life and social simulation, can provide a way forward for demographers who seek to escape the 'hungry beast' of highly data-driven research. This approach allows us to create models which develop a new understanding of population change without undue dependence on excessive empirical data, and create an environment in which models can be continually tweaked and worked on as new information comes to light, rather than simply sitting in stasis until the next wave of surveys comes back.

In the next section we will briefly outline the current state-of-the-art in demography, with focus on the contemporary limits of demographic knowledge.

## Where the Beast Lies: Demographic Knowledge and its Limits

Demography is currently facing major epistemological challenges. In particular, demographers' knowledge seems to have reached its limits with respect to the predictability of future population developments, as well as the ability to combine micro- and macro-level information and to find a compromise between the complexity and simplicity of analytical tools. This section discusses these issues in more detail.

The first problem with the limits of demographic knowledge is the issue of predictability. Amongst social science disciplines, demography has a unique predictive potential. Unlike in economics or sociology, very important information on the future development of populations is already embodied in their own age structures. The main mechanism of demographic dynamics is known, too: human populations change through births, deaths and, if considered at sub-global levels, migrations. However, when considered on their own, these three components of population change remain largely uncertain (Hajnal, 1955; Orrell, 2007). They also differ with respect to their degree of predictability: mortality is considered to be the best-predictable component; migration — the worst; fertility being usually located in the middle (National Research Council, 2000).

In the context of the uncertainty of forecasts, predictability limits have been extensively discussed elsewhere (Keyfitz 1981; Keilman's contribution to Willekens 1990; de Beer 2000; Bijak 2010), with two main methodological conclusions. Firstly, it is argued that demography should embrace uncertainty more closely (Alho and Spencer, 2005), in particular by moving from traditional deterministic projections to probabilistic forecasts. Secondly, there is an agreement that with longer horizons — beyond 10 to 20 years — uncertainty anyway becomes too large to be usefully described in probabilistic terms, and hence there is a need to turn to scenario-based approaches (see also Orrell and McSharry, 2009; Wright and Goodwin, 2009). An open question is, which elements should be included in such scenarios and how should they be constructed?

The second limitation of demographic knowledge stems from the problem of aggregation. Populations are composed of individuals and, as argued by Courgeau (2007), focusing exclusively on macro or micro-level analysis can generate problems with either ecological or atomistic fallacy. Whilst demography until the 1980s was almost entirely pre-occupied with the macro level, and since then increasingly more with the individual level (mainly in a form of the event-history analysis allowing for microsimulations), attempts to bridge both levels are much more recent (Willekens, 2005; Courgeau, 2007; Zinn et al., 2009). Microsimulations, as noted by Gilbert and Troitzsch (2005, p. 8), are predictive simulation tools "based on a large random sample of a population of individuals, [which are] 'aged' using a set

of transition probabilities [...], [so that] aggregate characteristics can be calculated and used as estimates of the future characteristics of the population". Micro-level simulation models, as well as their multi-level extensions, are usually also multi-state, states being for example age groups, educational classes, or states of health. In such models, individuals move between the states according to some transition probabilities, usually estimated on the basis of large-scale representative surveys, population registers or census data.

The main challenge with the multi-level approaches lies with their potentially enormous data requirements owing to the combinatorial explosion of the parameter space at different levels. That is exactly where the beast lies: Burch (2003) identified it to be the realm of logical empiricism, on which demography was — and still is — over-reliant. This philosophy focuses on observable phenomena and attempts to create generalisations solely on an empirical basis. As a result, in contemporary research problems driven by real-life questions concerning more complex phenomena, the beast can quickly become insatiable.

The third epistemological dilemma of contemporary demography stems directly from the previous two. At its core there is a question, whether complex models are more useful to aid prediction and decision making than their simpler counterparts. In terms of predictive performance, there is no evidence that complex models perform better (Ahlburg, 1995; Smith S.K., 1997). If that is the case, there might be a temptation to follow the Occam's razor principle (or the KISS principle in complexity science), disregard the additional subtleties involved in the modelling process and opt for simplicity instead (Bijak, 2010). However, such approaches may not increase our understanding of the underlying mechanisms, and are largely limited to shorter time horizons of decision making. To move beyond that, a different approach to modelling would be required.

From this perspective, the following section discusses the applicability of agent-based models in demography, with focus on how they could address the three challenges mentioned above.

### Agent-Based Demography: Avoiding the Beast

In their seminal book, Billari et al. (2003) present a compelling argument for the use of agent-based models in demography, or what they refer to as 'agent-based computational demography' ('ABCD'). Their enthusiasm for this approach stems from the potential for agent-based models to build theories regarding social processes that underlie demographic change. They describe a new ethos for simulation in demography, in which "the simulation is used first of all to develop and explore theories rather than to evaluate empirically the consequences of given rates/probabilities" (Billari et al., 2003, p. 11).

The suitability of agent-based models for exploring theories is certainly attractive for social scientists, as such mod-

els are well-positioned to examine the link between individual behaviour and higher-level organisation (Silverman and Bryden, 2007). In demography, agent-based models provide a potential platform in which the dynamic relationship between the micro- and macro-levels of a simulated population can be more fully represented. While in recent years multi-level microsimulation models, such as the ones mentioned above, have become increasingly popular, these modelling platforms still fail to capture the influence of micro-level behaviour and agent heterogeneity on macro-level entities, and indeed the feedback of those entities on agent behaviour. Nor do they capture social interactions, formation of social networks, or other elements which may contribute to the social processes underlying demographic change — here, agent-based models are more suitable (Gilbert and Troitzsch, 2005).

Beyond these theoretical benefits, we propose that in the specific context of demography, agent-based modelling offers a possible means to escape some limitations to knowledge imposed by the currently dominant data-based methodological paradigm. The first limitation — the one of predictability — points us toward the potential for using agent-based models for generating scenarios, which would produce useful insights about demographic change over a longer time horizon. A great advantage of agent-based models lies in their suitability for exploring a set of scenarios based upon varied parameter settings. Modellers can develop such scenarios based on variations within a parameter space, which allow them to examine how these parameters affect agent behaviour (and, in appropriately designed models, how those behaviours affect macro-level entities). In the development process, boundaries to the scenarios are limited only by the modellers' imagination rather than by data availability alone.

The second challenge for demographers — the aggregation problem — again points toward agent-based models as a possible way forward. After all, some ambitious social simulations not only include individual agents, but may also include macro-level components and thus allow for feedbacks between individuals, as well as between micro- and macro-level (Billari et al., 2003; Murphy, 2003; Silverman and Bryden, 2007). This would allow the modeller to neatly side-step the problem of focusing exclusively on either the micro- or macro-level, providing an opportunity to evade either ecological or atomistic fallacy. Such models can also conceivably capture downward causation effects and other manifestations of links between the micro- and macro-level, which would be impossible in a model which focuses only on one level or the other. Of course, this second challenge also allows the beast to begin rearing its ugly head. As mentioned in the previous section, the prevalence of the logical empiricist approach in demography places a certain primacy on deriving sensible results *only* from empirical observation (Burch, 2003). This naturally leads demographers to seek

out ever larger and more comprehensive data sets, each more expensive and time-consuming to collect than the last.

We then find ourselves sat facing the third challenge — that of simplicity. The beast gets hungrier for more data, and the sets of numbers which need crunching continue to grow in response. Agent-based models, however, necessitate a different approach: *data* is given less primacy than *parameters*. Rather than extrapolating from a given dataset about a population, social simulations will attempt to *generate* a society using the given parameters. The latter can certainly be informed by real-world data whenever they are available.

Thus, a type of modelling used quite often to represent complex systems might require less numerical data input than traditional methods. In certain contexts, social scientists may not find any data necessary at all — as in Schelling (1978), which demonstrated a possible mechanism for residential segregation based on individual behaviour without requiring any data, and only using a single parameter. Notably, Schelling was able to achieve this by focusing exclusively on a possible mechanism for residential segregation, and did not seek any relationship to empirical data; models for demography would need to have some connection to data to remain connected with real-world data and retain some potential predictive capacity. However, Schelling's model is a useful example in that it shows the potential for reducing the need for a model to be *entirely dependent* on data.

As an additional benefit, agent-based models can more sensibly be informed by qualitative data than traditional demographic modelling methods. Such data often explicitly attempts to “elicit agent models directly rather than inferring them from behavior” (Chattoe, 2003, p. 52).

So, agent-based models can present demographers with a way to avoid the beast and get away from ravenous traditional models which require regular feedings of painstakingly collected data. However, using such models in demography may require a certain shift in focus: agent-based models are better-suited for exploring theories and scenarios than for making firm predictions (Epstein, 2008). Therefore, perhaps we may take inspiration from John Hajnal — himself one of the most prominent demographers of the 20th century — and focus on building models which “involve less computation and more cognition than has generally been applied” (1955, p. 321). In this context, we understand the terms ‘computation’ and ‘cognition’ in the spirit of Hajnal's paper — the former strictly related to data-based predictions, and the latter to the explanation of the underlying demographic phenomena.

As we shall see in the following section, attempts to bolster the power of traditional data-driven models have not always been successful — and agent-based models have already been proven useful in some areas of demography.

## Analysis: Case Studies of Demographic Models

In demography, there are notable examples of models that fell short of their proclaimed aims due to the presence of the data-hungry beast. With respect to approaches spanning the micro and macro levels, an interesting attempt to apply methods from the system dynamics tradition to a demographic problem — migration — was the one by Weidlich and Haag (1988). Their approach was rooted in theoretical physics, in particular thermodynamics, and involved the estimation of individual-level transition rates between different regions (states of the system), based on the construed utility function of individuals and a set of macro-level covariates describing the regions. These quantities were linked through a set of *master equations* — first-order differential equations, describing the probabilities of the whole system moving from one state to another following the relocation of individuals. However, solutions proposed by Weidlich and Haag (1988), despite their mathematical sophistication and elegance, did not become a part of demographers' toolkit. There were several reasons for this. Some reviews of Weidlich's and Haag's book stressed that their method did not take into account heterogeneity of migration with respect to age, sex and past migration history<sup>1</sup>. Other points of criticism were that the approach did not model agents at all, thus not exploring the underlying social complexity in full, and did not provide many examples of empirical applications, mainly due to very large data requirements<sup>2</sup>. Finally, the quasi-deterministic nature of the models made them overly reliant on analytical solutions to the system of differential equations describing the dynamics of the migration system in question.

More recently, the MicMac project, as previously mentioned, aimed to develop a new methodology for dynamic microsimulation in demography (Willekens, 2005; Zinn et al., 2009). The final MicMac model consists of a macro-level part, which examines demographic change at the population level with a top-level macrosimulation (known as *Mac*), together with a dynamic microsimulation model that examines demographic events at the individual level (known as *Mic*). Both components of MicMac generate projections based on transitions between demographic states, but *Mac* generates cohort biographies while *Mic* generates individual biographies. In this way the model aims to bridge the micro-macro gap, providing a comprehensive modelling package which can pinpoint the influences of micro-level demographic events on macro-level demographic change (see also Billari et al., 2006).

<sup>1</sup>Daniel Courgeau's review of Weidlich and Haag (1988). *Population* 46(5), 1991: 1298–1299.

<sup>2</sup>J. Barkley Rosser's review of W. Weidlich's (2002) book "Sociodynamics: A Systematic Approach to Mathematical Modelling in the Social Sciences." *Discrete Dynamics in Nature and Society*, 3, 2005: 331–335.

In practice, however, the beast once again rears its head, and data requirements in this case are substantial. The microsimulation portion of the model (*Mic*) requires a significant amount of detailed micro-level data to implement, especially on transition rates between all possible demographic states for individuals<sup>3</sup>. The macro-level model (*Mac*) also requires extensive data about transition rates in order to run. Given that *Mic* includes 12 variables for each individual, very large amounts of input data are required to produce age- and time-specific transition rates between all possible states.

In turn, from the opposite — agent-based — end of the modelling spectrum, one example of an agent-based model producing some historical demographic insight is the model of the Kayenta Anasazi civilisation (Axtell et al., 2002). The model attempts to explain the rapid decline of the Kayenta Anasazi tribe in Long House Valley in northeastern Arizona, United States. The Anasazi tradition began in the area around 1800 B.C., when maize was introduced as a major agricultural crop. Around 1300 A.D., the population declined rapidly, and eventually there was a mass exodus from the valley.

The model of Axtell et al. (2002) consists of a digital reconstruction of the Long House Valley landscape, constructed using existing knowledge of the environmental conditions at that period in history. The agents themselves represent households, individual people being more difficult to identify with any reliability using the existing archaeological data. Each household has certain rules of behaviour which specify how it will select its dwelling and planting locations during each calendar year based on how successful it has been at satisfying its nutritional needs.

The model seemed to produce a simulated population which closely followed the ebbs and flows of the real Anasazi population in Long House Valley. Interestingly, however, the model shows that some small sustainable population could have remained in the northern part of the valley, even as the environmental conditions started to degrade toward 1300 A.D.; this contrasts with the real population, in which the remaining people joined the mass exodus leaving the valley.

This model thus demonstrates that the demographic changes which affected the Anasazi population in this area can be explained at least in part by an agent-based model with simple behavioural rules. As the environment degrades over time, and the agents must continue to look for fertile ground in which to plant their fields, the simulated population shifts northward, just as the real Anasazi had done. In contrast, a demographic model which did not capture these rules of individual behaviour may have been able to accurately portray the changes occurring at an aggregate level,

<sup>3</sup>See Deliverable D9 of MicMac: "Report on Data Requirements of MIC" by F Willekens, J de Beer and N van der Gaag: <http://www.nidi.knaw.nl/Content/NIDI/output/micmac/micmac-d9.pdf>

but would not be able to explain *why* those changes occurred.

Of course one might ask, how did this model keep the beast from getting out of control? The model clearly incorporated many pieces of information from a variety of disciplines. However, it is interesting to note in this case *how* the beast was fed. Interestingly from a demographic point of view, along with already-available archaeological information, the model was able to incorporate *qualitative* data in the form of ethnographical research: the agents' behavioural rules were formulated by distilling ethnographic knowledge about the Anasazi civilisation into simple rules driving their migration and agricultural activities.

Another example of an agent-based model producing demographic insight is a recent study of marriage offered by Billari et al. (2007). Their model was constructed as an attempt to bridge the gap between two different perspectives which predominate in the study of the timing of marriage in populations: macro-level statistical modelling used by demographers, and micro-level studies performed by psychologists and economists examining the partner (mate) search process. In this context, an agent-based model is seen as a possible way to "account for macro-level marriage patterns while starting from plausible micro-level assumptions" (Billari et al., 2007, p. 60).

The resulting model assumes that the formation of marriage partnerships is the result of social interaction between heterogeneous agents. The model attempts to demonstrate the link between these interactions and marriage patterns by simulating the impact of the availability of mates and the desirability of marriage, which is affected by the influence of relevant others in an agent's social network. The results show that the model can reproduce the hazard functions of marriage observed at the population level in the real world. The performed sensitivity analysis suggests that the results are robust to changes in the relevant simulation parameters.

The findings of Billari et al. (2007) have important implications for demographers wishing to avoid the beast. As the authors note, the model uses substantial simplifying assumptions: placing the agents in a one-dimensional, circular space; leaving out additional social complexities such as courtship or divorce; and focusing only on age and location as agent attributes, ignoring kinship, education, occupation, socio-economic status, or other similar factors. In fact, the simulation almost entirely ignores any empirical data, with the exception of the initial population which is generated with an age distribution reminiscent of 1950s America.

Despite the paucity of data, however, the simulation-based demographic models seem to produce at least plausible micro-level explanations of macro-level phenomena. In the work of Billari et al. (2007) this concerns the influence of social pressure to get married within a social network, and the variation of the size of that network by age is a determinant of the desirability of marriage. In contrast, a macro-level statistical model of marriage timing would not

be able to provide this sort of micro-level explanation — and would require significantly larger investments into data collection in order to function. In turn, the study of Axtell et al. (2002) captured the main factors behind the expansion and twilight of the Anasazi population. On the other hand, even painstaking efforts to reconstruct birth, death and migration rates based on fragmented pieces of historical information would yield demographic predictions that would be too uncertain to be meaningful, if only the levels of uncertainty were honestly admitted in such a model. The beast might be fed and sated — but our understanding of the underlying processes would be no greater.

## Conclusions

Our discussion and analysis have demonstrated that traditional demographic methods, while highly accomplished in producing data-driven population projections, face some major epistemological and pragmatic challenges. The overall focus on data over theoretical investigation has hampered demography's ability to provide explanations of demographic change, while the hunger of the beast of logical empiricism traps demographers in continuous cycles of expensive and time-consuming data-collection.

As we have seen, the application of agent-based methods inspired by contemporary ALife work to demography provides a means to lessen some of these burdens on population researchers. The resultant increased focus on explanation over producing projections from empirical data could allow demographers to develop more coherent micro-level explanations of macro-level demographic change. More pragmatically, the concomitant reduction in data dependence would reduce the hunger of the beast, allowing demographers more freedom to produce varied and ambitious models while also removing the restrictive timetables imposed by lengthy and expensive data-collection processes.

So far, all applications of agent-based models to population change, such as the ones mentioned earlier in our analysis, have been performed separately, abstracting away from the main mechanism and inertia of population dynamics. In particular, existing studies deal with particular components of population change — fertility and marriage (Murphy, 2003; Billari et al., 2007), or migration and residential patterns (Heiland, 2003; Benenson et al., 2003) — separately, without putting them together into a common modelling framework describing the known features of population dynamics. In our view, developing an integrated, multi-level and multi-state agent-based model could overcome some of the philosophical difficulties discussed before, whilst remaining related to the real-world through the empirical data, wherever they are already available. The challenge ahead is thus to build models which would combine various features of demographic processes and yield artificial populations equipped with real-world characteristics. In that respect, agent-based demography is not only interesting as a



research field, but also as a promising venue for answering questions relevant to policy makers. Moreover, it provides the users of the final research output with more possibilities for interacting with the researchers, by engaging in the experimentation with the artificial worlds created. For both parties involved in the process — researchers, as well as the end-users of research — this can bring about a better understanding of the underlying population processes, which itself can be a very important gain from the whole modelling exercise.

From these points of view, agent-based demography seems to be an innovative way of moving the whole research field in a new direction, towards the middle ground on the theory-data spectrum. For the ALife community, this ‘dialectic’ position would open up a whole new, fascinating field of research with direct applications to real-world problems. However, building agent-based models to population questions would require that demographers use more imagination than in pure data-driven modelling, in line with Hajnal’s *credo*. Agent-based modelling can offer a solution, which has to be based on cognition and thinking about mechanisms (Burch, 2003; Chattoe, 2003), while taking into account these pieces of information (data) that are already available. In that respect, the rule of the thumb for agent-based demographers who would like to strike the delicate balance between empiricism and explanation is: we should feed the beast where feasible — but not more.

## Acknowledgements

This work was supported by the EPSRC grant EP/H021698/1 “Care Life Cycle”, funded within the “Complexity Science in the Real World” theme. The authors are grateful to an anonymous ECAL reviewer and to Stuart Rossiter for comments which helped improve an earlier draft.

## References

- Ahlburg, D.A. (1995). Simple versus complex models: Evaluation, accuracy and combining. *Mathematical Population Studies*, 5(3): 281–290.
- Alho, J. M. and Spencer, B. D. (2005). *Statistical Demography and Forecasting*. Berlin–Heidelberg: Springer.
- Axelrod, R. (1984). *The Evolution of Cooperation*, New York: Basic Books.
- Axtell, R.L., Epstein, J.M., Dean, J.S., Gumerman, G.J., Swedlund, A.C., Harberger, J., Chakravarty, S., Hammond, R., Parker, J., and Parker, M. (2002). Population Growth and Collapse in a Multi-Agent Model of the Kayenta Anasazi in Long House Valley. *Proceedings of the National Academy of Sciences*, Colloquium 99(3): 7275–79.
- de Beer, J. (2000). Dealing with uncertainty in population forecasting. *Working Paper of the Department of Population Statistics*. Voorburg: Central Bureau of Statistics.
- Benenson, I., Omer, I., and Hatna, E. (2003). Agent-Based Modeling of Householders’ Migration Behaviour and Its Consequences. In: Billari F and Prskawetz A (eds.), op. cit. (pp. 97–115).
- Bijak, J. (2010). Forecasting International Migration in Europe: A Bayesian View. *Springer Series on Demographic Methods and Population Analysis*, vol. 24. Dordrecht: Springer.
- Billari F., Aparicio Diaz B., Fent T. and Prskawetz A. (2007). The “Wedding–Ring”. An agent–based marriage model based on social interaction. *Demographic Research*, 17(3): 59–82. <http://www.demographic-research.org/Volumes/Vol17/3/17-3.pdf>
- Billari F., Fent T., Prskawetz A. and Scheffran J. (eds.) (2006). *Agent–Based Computational Modelling: Applications in Demography, Social, Economic and Environmental Sciences*. Heidelberg: Physica Verlag.
- Billari F.C., Ongaro F. and Prskawetz A. (2003). Introduction: Agent–Based Computational Demography. In: Billari F and Prskawetz A (eds.), *Agent–Based Computational Demography: Using Simulation to Improve Our Understanding of Demographic Behaviour*. Heidelberg: Physica Verlag (pp. 1–17).
- Billari, F.C. (2006). Bridging the gap between micro–demography and macro–demography. In: G. Graziella, J. Vallin and G. Wunsch eds. *Demography: analysis and synthesis*, Vol 4. New York: Academic Press (Elsevier) (pp. 695707).
- Burch, T.K. (2003). Data, Models, Theory and Reality: The Structure of Demographic Knowledge. In: Billari F and Prskawetz A (eds.), op. cit. (pp. 19–40).
- Cederman, L.–E. (1997). *Emergent Actors in World Politics: How States and Nations Develop and Dissolve*. Princeton, NJ: Princeton University Press.
- Chattoe, E. (2003). The Role of Agent–Based Models in Demographic Explanation. In: Billari F and Prskawetz A (eds.), op. cit. (pp. 41–54).
- Courgeau, D. (2007). Multilevel Synthesis: From the Group to the Individual. *Springer Series on Demographic Methods and Population Analysis*, vol. 18. Dordrecht: Springer.
- Di Paolo, E. A., Noble, J., and Bullock, S. (2000). Simulation models as opaque thought experiments. In M. A. Bedau, J. S. McCaskill, N. H. Packard, and S. Rasmussen (Eds.), *Artificial Life VII: The Seventh International Conference on the Simulation and Synthesis of Living Systems* (pp. 497–506). Cambridge, MA: MIT Press/Bradford Books.
- Epstein, J. M. (2008). Why model? *Journal of Artificial Societies and Social Simulation*, 11(4)–12.
- Gilbert, N. and Troitzsch, K.G. (2005). *Simulation for the Social Scientist*. 2nd ed. Maidenhead: Open University Press.
- Hajnal, J. (1955). The prospects for population forecasts. *Journal of the American Statistical Association*, 50(270): 309–322.
- Heiland, F. (2003). The Collapse of the Berlin Wall: Simulating State-Level East to West German Migration Patterns. In: Billari F and Prskawetz A (eds.), op. cit. (pp. 73–96).

- Keyfitz, N. (1981). The limits of population forecasting. *Population and Development Review*, 7(4): 579–593.
- Langton, C.G. (1989). Artificial life. In: Langton C G (ed.), *Proceedings of the Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (ALIFE '87)*, Redwood City, CA: Addison–Wesley (pp. 1–48).
- Lem, S. (1983). Pożytek ze smoka. Reprinted in: (2008) *Kongres futurologiczny. Opowiadania Ijona Tichego*. Warsaw: Agora (pp. 179–187). [Own translation – JB]
- Murphy, M. (2003). Bringing Behavior Back into Micro–Simulation: Feedback Mechanisms in Demographic Models. In: Billari F and Prskawetz A (eds.), op. cit. (pp. 159–174).
- NRC [National Research Council], (2000). *Beyond six billion: Forecasting the world's population*. Washington, DC: National Academies Press.
- Orrell, D. (2007). *The Future of Everything: The Science of Prediction*. New York: Thunders Mouth Press.
- Orrell, D. and McSharry, P. (2009). System economics: Overcoming the pitfalls of forecasting models via a multidisciplinary approach. *International Journal of Forecasting*, 25(4): 734–743.
- Rossiter S., Noble J. and Bell K.R.W.(2010). Social simulations: Improving interdisciplinary understanding of scientific positioning and validity. *Journal of Artificial Societies and Social Simulation*, 13(1)–10.
- Sawyer, R.K.(2005). *Social emergence: Societies as complex systems*. New York: Cambridge University Press.
- Schelling, T. (1978). *Micromotives and Macrobehavior*. Norton, New York, USA.
- Silverman, E. and Bullock, S. (2004). Empiricism in Artificial Life. In *Artificial Life IX: Proceedings of the Ninth International Conference*, MIT Press: Cambridge, MA, USA (pp. 534–539).
- Silverman, E. and Bryden, J. (2007). From artificial societies to new social science theory. In Almeida e Costa F, Rocha L M, Costa E, Harvey I and Coutinho A (eds.) *Advances in Artificial Life, 9th European Conference, ECAL 2007 Proceedings*. Berlin–Heidelberg: Springer (pp. 645–654).
- Smith, S.K. (1997). Further thoughts on simplicity and complexity in population projection models. *International Journal of Forecasting*, 13(4): 557–565.
- Smith, V. (2008). Evolving an agent–based model to probe behavioral rules in flocks of cowbirds. In Bullock S, Noble J, Watson R and Bedau M A (eds.), *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*. Cambridge, MA: MIT Press (pp. 561–568).
- Toquenaga, Y., Kajitani, I., and Hoshino, T. (1995). Egrets of a feather flock together. *Artificial Life*, 1(4): 391–411.
- Weidlich, W. and Haag, G. (eds.) (1988). *Interregional migration: Dynamic theory and comparative analysis*. Berlin–Heidelberg: Springer.
- Willekens, F. (1990). Demographic forecasting: state–of–the–art and research needs. In CA Hazeu and GAB Frinking (eds.), *Emerging issues in demographic research*. Amsterdam: Elsevier (pp. 9–66).
- Willekens, F. (2005). Biographic forecasting: bridging the micro–macro gap in population forecasting. *New Zealand Population Review*, 31(1): 77–124.
- Wright, G. and Goodwin, P. (2009). Decision making and planning under low levels of predictability: Enhancing the scenario method. *International Journal of Forecasting*, 25(4): 813–825.
- Zinn, S., Gampe, J., Himmelspace, J. and Uhrmacher, A. M. (2009). MIC–CORE: a tool for microsimulation. In Rossetti M D, Hill R R, Johansson B, Dunkin A and Ingalls R G (eds.) *Proceedings of the 2009 Winter Simulation Conference*. IEEE (pp. 992–1002). <http://ieeexplore.ieee.org>

# A Schematic Representation of Autopoiesis Using a New Kind of Discrete Spatial Automaton

Jean Sirmai<sup>1</sup>

<sup>1</sup>Clinician and Hospital Medical Practitioner, Ile de France.  
jean.sirmai@orange.fr

## Abstract

Autopoiesis, or one's ability to renew oneself, is a meaningful concept for the study of life. Modeling of autopoiesis would enhance its study in relation with other biological properties such as feeding, breeding, being ill, healing, dying, etc. Here, we report the design of a "morphautomaton", a new kind of discrete spatial automaton designed to represent within the same space an unlimited number of various complex, mobile, interacting forms. This automaton uses a simple, single effective formalism to identify and localise those forms and describe their movements, transports, and transformations. We make use here of these forms as symbols to schematically represent an autopoietic individual within its environment. This representation can be made consistent with the laws of thermodynamics and conservation. Using this representation, the study of the physiological properties of this individual could be undertaken. Using this platform, the modeling of other biological properties in relation with autopoiesis should also be possible. These models should allow future comparisons, definition, and classification of these biological properties. Representations using our formalism and similar parameters can be combined. Because it focuses on the physiological analysis of whole individuals, this schematic representation method can be used only when structural analysis has been completed.

## Introduction

The word "autopoiesis" was created by Francisco Varela and Humberto Maturana in 1971 to designate the ability of something to "create oneself" (Maturana and Varela, 1973). The ability of an individual to renew itself while maintaining its shape and organization is remarkable. Indeed, no inanimate objects have such a property, yet living objects often heal and return naturally to their original forms. How do they accomplish this? Where does the difference arise? Could we acquire control of it?

At one time, the idea that living matter differed from inanimate matter seemed obvious. Today, we have identified most parts that comprise living beings, and there is no doubt that these parts are made of the same material as inanimate objects (Goodsell, 2009). While anatomic knowledge is not sufficient to understand the differences between living and non-living, we can hypothesize that the set of physiological processes performed by these parts give the whole its properties (Schoenheimer, 1942; Schrödinger, 1944; Kleiber, 1947). Thus, here we search within this paradigm for a mechanical explanation of autopoiesis.

We remain far from understanding what roles each piece plays in creating the whole; neither can we perfectly analyze or reconstruct the simplest organisms. We know not what assumptions are necessary to guide such a reconstruction; what we need is not simply a technique but a method or theory providing the guidelines to direct the analysis and representation of our observations. Critical to this theory are the criteria that will enable a strict definition of the properties of the living: feeding, breeding, growing, being ill, healing, dying, evolving, behaving, etc. By removing or modifying some structures and observing changes in the whole, we can distinguish which structures are necessary and sufficient for the existence of these properties. By comparing several living entities, we can acquire an indication of their generality (applying to cell, organism, society) and of their relationships (dependence, anteriority, causality, equivalence, etc.). Indeed, we may go further and study the forms of these entities, to determine whether they can be classified and systematically enumerated, if they derive from each other, and how to most simply represent them.

However, complex systems have unpredictable emergent properties. To gain control over any property of such a system, one must forego, at least initially, studying what might emerge from it, as a system is either controlled or has emergent properties (Liu et al., 2011). Our goal here is to avoid the apparition of any emergent property, while creating an autopoietic representation that may lead to its control. Meanwhile, we wish to retain the ability to later study autopoiesis' associations with division and differentiation to produce systems endowed with the ability to evolve.

To define and control a property, two approaches are possible: synthetic, or "bottom-up" (used here), identifies the property then searches for a mechanism to reproduce it. This kind of approach has been infrequently explored because of several inherent weaknesses, such as its generality, the arbitrary choice of entities to represent the property, or the lack of analysis of physical constraints (Morange, 2005; Atlan, 2011). The majority of modeling works are analytical, "top-down" approaches: beginning with available experimental data, one selects useful parts and tries to reassemble them to reproduce a property of interest. The two approaches, however, are complementary and are needed to validate one another; when these approaches validate the same mechanism, the property becomes correctly established; its full control (i.e., the ability to calculate, reproduce, modulate,

or combine it with others in any particular physical, chemical, or even robotic context) can be envisioned.

To demonstrate autopoiesis, Varela and Maturana worked in a non-experimental, theoretical framework and built a dynamic representation *in silico* by programming a discrete spatial automaton to represent the minimal organization of a biological cell (McMullin and Varela, 1997). They intended to represent a membrane constantly and permanently destroyed and rebuilt through the actions of particles they called substrates, catalysts, and links. The destruction of this membrane was done at random and was offset by new syntheses. This first model was a great launching point for later works (Zeleny, 1977; McMullin and Gross, 2001; Ikegami and Suzuki, 2008; Bersini, 2010). However, Varela's model contains some weaknesses: The definition of autopoiesis (Maturana and Varela, 1980) is not clear and simple and is not fully applied in his later models, as destruction of the membrane, renewal of the catalyst, transmembrane transports, and, therefore, control of the individual's size are not performed by the individual itself. Varela did not use a well-formalized representation platform, but he worked in a time where it was difficult to separate the representation method from the represented object. Most of these works are restricted to chemical materials, and do not consider other potential implementations (e.g., robotic).

To move the study of autopoiesis forward, we propose to reformulate the definition of the property as follows: first, because autopoiesis is dynamic, we may hypothesize that it is a property of a whole that cannot be a unique, static, inanimate block, but that consists necessarily of distinct, interacting parts. To interact, these parts must be mobile. If they are mobile and yet they continue to be part of the whole, then they are joined. What connects them is also an aspect of the whole. The interactions that they continually have renew them. This means each of them can be destroyed by one and re-built by another. Importantly, if all the necessary conditions are met, then the whole can keep its form longer than the parts composing it. Because of continuous movement of its parts and exchanges with its environment, this form and its composition can never be exactly the same; rather, they fluctuate around a mean shape. Indeed, the whole loses this property if it is split or isolated from its surrounding environment, with which it exchanges energy and matter to achieve its renewal. Additionally, if there is a "natural" process of degradation of some of its parts, the whole's renewal speed should remain sufficient to oppose the effect of this process. Importantly, "autopoiesis" should also satisfy McMullin's criterion, which states that two individuals, in direct contact with each other, can reliably maintain their separate identities (McMullin, 2004).

The methods applied to the study of autopoiesis evolved in parallel of the concept itself. Early studies conducted by Tibor Ganti, Robert Rosen, Victor Kunin, Manfred Eigen, and others identified related concepts, but were generally performed using differential equations (Popa, 2004). However, differential equations, developed to describe quantitatively determined and continuous phenomena, assume ideally mixed reactants and are poorly adapted to molecular biology. Instead, discrete spatial automata can associate

logical operations to possibly non-linear and probabilistic numerical calculations and can apply them to irregular and discontinuous distributions of matter (Rao et al., 2002; Broderick et al., 2005; Wishart et al., 2005; Morris et al., 2010). Here we designed a particular new class of discrete spatial automata that can represent a set of complex, mobile, interacting forms whose evolution may schematically represent a phenomenon of interest; we use it to represent an autopoietic individual.

## Methods: Description of the morphautomaton

### Workspace, states, and transition rules

The *workspace* in this automaton consists of a set of regularly arranged, adjacent *tiles*<sup>1</sup>. Each tile has a *state*, which is described as either empty or occupied. A tile is occupied, and thus called a *particle*, when it has at least one *link*. Conversely, an empty tile has no link. By definition, two particles cannot occupy a single tile.

A link represents a directional association between particles. It belongs to one particle and indicates the position of a neighboring one<sup>2</sup>. The state description of the workspace is complete when each tile is known to be either occupied or empty, and, for each occupied tile, the orientation of each link is known. All information required to describe a workspace state can be acquired using only two distinct kinds of *conditions*: one refers to the number of links of a particle (isotropic conditions) and the other the orientation of a particle's links (anisotropic conditions). An isotropic condition contains coordinates for a tile and a number, *n*. The number of links in the indicated tile is evaluated for equivalence to *n*. Further, this condition helps determine whether the tile is empty: empty space is defined by a number of links equal to zero. A positive anisotropic condition contains coordinates for a tile and an indicator of orientation, *p*. This condition assesses whether a link from the selected particle is oriented to *p*. Because of the possibility of superposition (i.e., a particle may have multiple links in any given orientation), a built-in mechanism considers whether there are superposed links. A negative anisotropic condition contains the tile coordinates and *p*, as above, and requires that no link is oriented toward *p* at the specified location.

Each *rule* comprises a set of conditions that, when met, initiates *operations* producing a transition from one state to another (see Figure 1).

---

<sup>1</sup> We avoid the word "cell" because of potential ambiguity with the biological meaning.

<sup>2</sup> Typically, any tile indicated by a link is considered occupied; however, this restriction can be overcome: individual disconnected particles moving at random and containing an available link may then randomly attach to any aggregate. The connection will result in a change in shape and, potentially, identity of the aggregate. Thus, these disconnected particles could be used for representation of damage occurring to various materials, in some cases acting as a sort of "mutagen."

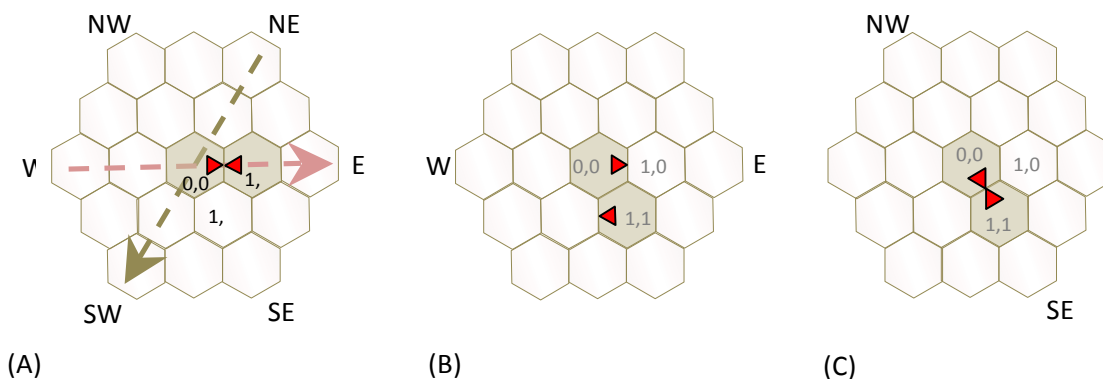


Figure 1. Example of a transition rule that describes simultaneous state change of two particles. A) The workspace above shows the neighborhood of the central cell (0,0). The space contains two dimensions, hexagonal tiling, and two non-orthogonal axes traveling west to east (x) and northeast to southwest (y). The central cell, a particle, contains a link (symbolized by the small triangle) facing east. East of (0,0) is another particle in (1,0), also containing a single link, this time oriented toward the west. The tile below, in (1,1), is empty. The states of other tiles are indifferent; these do not prevent or facilitate the application of a rule whose conditions permit identification of this situation. The following five conditions exist, whose order of evaluation is irrelevant: (a) number of links (0,0) = 1; (b) number of links (1,0) = 1; (c) number of links (1,1) = 0; (d) link (0,0) = East; (e) link (1,0) = West. In our example, all conditions are met, so the rule identifies the situation and thus can be applied. Several operations are then performed in a specific order: The tiles located in (1,0) and (1,1) are swapped using the operation [move]. The result of this translation is shown in B. Two [turn] operations then change the orientation of links of particles (0,0) and (1,1), which produces the result shown in C.

A rule must contain at least one condition and one operation. Possible operations are either *link rotations* or *particle translations*. Link rotations re-orient one link of a tile from one direction to another. Particle translations exchange the contents of two tiles, adjacent or not. If one of the two tiles is empty and the other is occupied, a simple particle translation occurs. However, if both tiles are occupied, a double translation or permutation occurs. The numbers of links and particles are preserved. The number of links in a particle does not change.

The conditions and operations of a rule do not apply to or affect the entire workspace, but only a particle and its local neighborhood comprising the tiles of interest. Rules must refer to the coordinates of the neighborhood to be evaluated and enacted. A selected particle represents the center of the neighborhood, and one of its randomly chosen links provides a reference orientation for this space. These selections define and orient the neighborhood. The choice of the central particle is made at random, and the automaton guarantees that all particles can be selected and none excluded. Each cycle of the automaton evaluates the same rule set in a different neighborhood. The cycle begins upon selection of a particle and one of its links, delimiting and orienting the neighborhood. Rules are then evaluated individually and sequentially<sup>3</sup>. If any one condition is not satisfied, that rule is rejected and the following one is evaluated. If all conditions

attached to a rule are met, then all operations of this rule are performed and the cycle ends. If all rules have been evaluated and none applies, then no state change occurs during this cycle. Once a cycle is completed a new cycle begins. During the evaluation of rules, each tile in the neighborhood can be evaluated for one or more conditions, and, when a rule has been satisfied, can be involved in one or more transitions. Importantly, tiles located outside the neighborhood remain unaffected by this process.

In summary, the workspace contains particles of one-unit size that can be linked and all operations on these particles and their links are performed locally<sup>4</sup>.

## Aggregates

*Aggregates* are groups of *particles* associated by *connections* (Figure 2). A particle is a full tile defined as having at least one link. Connections between particles are realized by the links between them. The simplest connection is achieved by one link between two particles; however, the number of links constituting a connection is unlimited. Links can also be superimposed (i.e., there may be multiple links in either direction between two adjacent particles). Additionally, the

<sup>3</sup> An important feature of this automaton is the addition of a random condition to decide between rules with identical local conditions. Sequential evaluation would normally cause the first of these rules to be applied and those following to be always ignored. The addition of a non-local random condition enables all concerned rules to be applied with equal probability. While rarely used, the random condition is essential for reproducing the randomness of some movements.

<sup>4</sup> Multiple threads can work in parallel and asynchronously on the same workspace. This feature requires that each neighborhood is treated separately, thus no interference exists between them. Whether a single thread operates or multiple threads are simultaneously active, the system's history is constructed in random order. Thus, two successive experiments do not follow the same trajectory. Reproducible results therefore testify to the independence of the model in relation to the mechanics of the automaton.

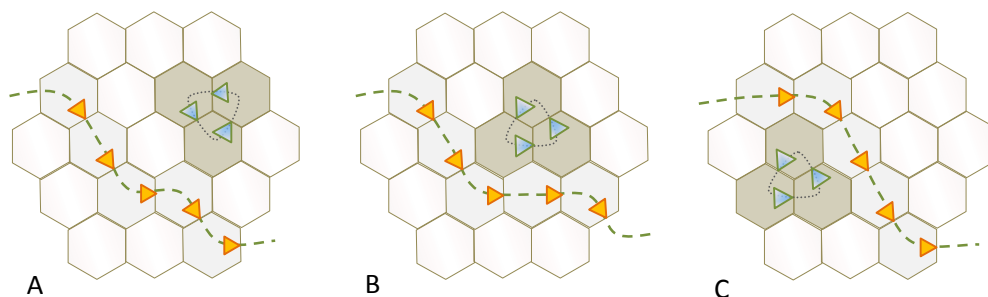


Figure 2. Two aggregates in motion. One aggregate is a grouping of three particles (a "trimer") and the other is a "chain" that is not entirely visible in the neighborhood (dashed line added for visibility); the chain delimits two compartments. As the chain's connections are asymmetrical, their orientation allows local identification of each of these compartments. Different movements apply to these two aggregates. The trimer, fully visible in the neighborhood, can be moved around by the action of a single rule; the long chain can only be distorted by any single rule. However, many successive distortions, performed at different locations, give the chain a full mobility in all directions. The transition from (A) to (B) does not alter the composition of the compartments, but the transition from (B) to (C), a transport, changes this composition because the trimer passes from one compartment into the other.

connection is called "reciprocal" when each particle in a connection has at least one link pointing to its connected neighbor. It is called "symmetrical" if the two particles run the same number of links to one another. Symmetrical connections are necessarily reciprocal.

The smallest aggregate is a dimer, composed of two particles. Aggregate size is not limited by the tiling used or the automaton mechanism, but may be limited by the dimensionality and the extent of the available space. Further, aggregates can be of any regular or irregular shape: monomer, polymer, chain, ring, bifurcation, helix, knot, cavity, grain, rotor, stator, etc. A good choice of forms and rules can produce representations of many material properties: hardness, flexibility, elasticity, rigidity, fluidity, permeability, etc. These lists are not exhaustive. An aggregate's size or the number and extent of its links are infinite. Further, basic forms can be disjointed or contiguous and combined in various ways. These features mean that this formalism provides open-ended possibilities of creating and combining elementary forms.

The operations performed on aggregates are associations of the basic operations performed by the rules. They can produce *movements* (displacement or distortion), *transformations*, or *transports* (movements from one compartment to another).

Movements can be translations, rotations, or combinations of both, and can be performed on the whole or a specifically identifiable part. They occur easily when the aggregate size is smaller than the neighborhood. When the aggregate size exceeds that of the neighborhood, however, it cannot be moved in its entirety by executing a single rule. However, such aggregates may be distorted. Carried out in multiple locations and repeated, distortions give large aggregates their mobility (Figure 2). The size of the neighborhood can be chosen according to the mechanical properties of the aggregates whose movements are to be represented. Each aggregate has an identity comprised of morphological local traits resulting from the unique pattern of its connections. The number of identifying properties is unrestricted and each identity can be evaluated by rule conditions. For example, one chain may have a linear structure and asymmetric connections of just one link, while

another chain may have symmetric connections, and another, asymmetric connections but with two or three links. As movements translate particles and reorient their links but never break connections, they always preserve the unique identity of each aggregate. By contrast, transformations always break at least one connection and reconnect some particles in another way, modifying aggregate identities. Transports preserve aggregate identities but move them from one compartment to another.

In summary, 1) aggregates are groups of connected particles of various forms whose size is not limited; 2) each one has a specific connections pattern that enables its identification; 3) aggregates are localised (in a compartment or as a limit of a compartment); 4) they can be moved, transported, or transformed; 5) transformations and transports, but not movements, require aggregate identification; 6) movements conserve identity and localisation; 7) transformations break and re-establish links, modifying aggregate identity; and 8) transports respect their identity but modify their localisation.

## Summary

The new type of discrete spatial automaton, which we call a "morph-automaton," (from ancient Greek morphè: form) is relevant because it allows the representation by means of a simple, single formalism of many forms moving and interacting in the same space. Indeed, while this formalism does not limit the diversity of these forms or the variety of operations that can be performed on them, the use of a unique rule structure enables automatic handling of the rules (editing, classification, presentation, optimization).

A number of parameters can be adjusted without changing the principles that underlie morphautomata construction and design. These parameters primarily concern the general structure of the workspace: They define its number of dimensions and extent, the type of tiling used, the existence of edges and the representation of general strength fields (e.g., electromagnetic, gravitational). Some parameters concern the shape and size of the neighborhood. Others parameters include the number of links per particle and the scope of these



links—and, thus, the variety of possible connections. Adjustment of these parameters depends on the phenomena that the experimenter wishes to represent. Importantly, models with identical parameters can be combined. Such operations may be formalized.

The variety of aggregates is unlimited, and, even constrained by the necessity of using discrete forms and limited neighborhood size, proper selection of these forms and of the operations that apply to them can produce schematic representations of a variety of complex heterogeneous materials. The experimenter chooses some of these forms and details when editing the initial state. He selects and describes their movements and interactions when writing rules. These choices depend solely on his assumption and the result he wants to achieve: it is his intention that gives logical consistency and value to the writing of the whole. Further, the experimenter is responsible for verifying the logical coherence of elementary conditions and operations composing each rule and their coherency with other rules and the state.

If the rules associated with each aggregate allow movement in all directions, and the automaton uses a random selection of particles, then the motion of each aggregate, isolated or in relation with others, is permanent and random. Transformations, moreover, have some similarities with chemical reactions. Additionally, conservation and

thermodynamic laws can be represented. Thus, this platform makes possible a schematic representation of certain physical and chemical properties of macromolecules in solution.

## Results: Autopoiesis

### Anatomy of the autopoietic individual

Our automaton has been designed with the intent of representing an autopoietic individual moving at random within its environment. It makes use of five varieties of aggregates. First, we describe the initial state of our workspace for the positions of each instance of the five different types of aggregates used to build the autopoietic individual. This detailed and comprehensive description of the initial state is required by the automaton machinery for the representation to evolve, and it must be edited manually. Required characteristics are those that denote each aggregate's identity, shape, and localisation.

The individual is composed of a membrane enclosing an internal compartment (Figure 3). This membrane is a circular chain made of one-link particles each pointing at the next. Because of a unique sense of rotation, its inner and outer faces are locally identifiable, delimiting two compartments: internal

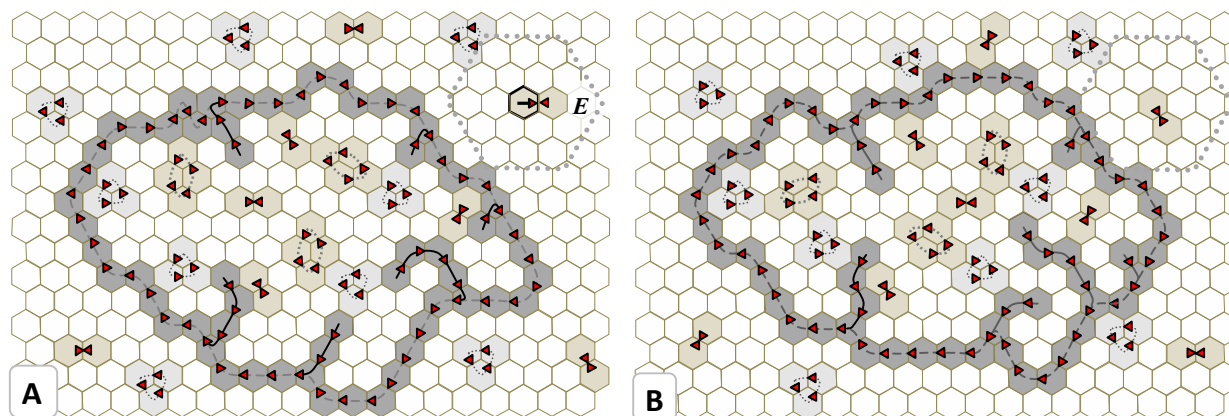


Figure 3. (A) and (B) represent the same autopoietic individual and its environment in two successive snapshots. Each panel represents the same part of the workspace; the two-dimensional space is covered by a hexagonal matrix whose tiles are either empty or occupied. Full tiles, called particles, have a link, symbolized by a small triangle, which refers to another particle; they are thus associated to form aggregates. One of these aggregates is a circular chain, called a membrane, which delimits two compartments: internal and external. In these compartments are several small isolated aggregates consisting of two, three, or four particles. Aggregates of four particles can be found in the interior compartment only. Several small chains ranging in length from one to four particles are attached at the internal face of the membrane.

The large hexagons, delimited by dots and located at the top right of each image, indicate the size of the neighborhood of a particle where evaluation of the rules occurs. (This part of the figure is superimposed with Figure 1A.) The particle at the center of this neighborhood was chosen randomly, and its link assigns the orientation during rules evaluation (black arrow, letter E see fig 1A). This particle is part of an aggregate consisting of two particles, called a dimer. (B) shows the new position of this aggregate after application of the rule.

Comparison of (A) and (B) shows the effect of many random movements of this type. The positions of most small aggregates have changed slightly. The membrane and small chains have been deformed and displaced, but their lengths are identical. The identity of each aggregate has been preserved, as no transformations occurred. The composition of each compartment was also preserved, as no transport occurred. Even if some traits of this abstract representation may evoke a living cell (which was the initial intention), it is definitely not one. This individual may only be an autopoietic representation: it has some very particular features that no real cell will ever have, and is not intended to represent other properties of real cells.



Table 1: Summary of operations performed by the automaton.

		Action	Conditions
Transformation	1	membrane > chain	Membrane can wrinkle, its continuity can be ensured, a chain at least one-unit length is already there
Transformation	2	chain > tetramer	Chain length is almost four units long
Transformation	3	tetramer > dimers	Other tetramers are in proximity
Transport	1	trimers entry	Enough space available, membrane flexible, no chains attached to the membrane locally
Transport	2	dimers exit	Enough space available, membrane flexible, no chains attached to the membrane locally
Transformation	4	trimer > membrane (one unit) + dimer	Only in the presence of a tetramer
Transformation	5	Trimer > new chain (one unit) + dimer	Only in the presence of another trimer
Transformation	6	two chains one unit each > dimer	Other one-unit chains in proximity

and external. Dimers, trimers, and tetramers are small aggregates, disconnected from others and made, respectively, of 2, 3, or 4 particles. As a particular characteristic of this model, tetramers are found only inside the internal compartment; dimers and trimers are both inside and out. On the inner side of the membrane may be attached small chains comprising 1, 2, 3, 4 or more one-link particles.

In summary, five varieties of aggregates are used in several instances in this representation: membrane, dimer, trimer, tetramer, and chain. Their relative localization and shapes are essential information.

### Physiology of the autopoietic individual

We must now focus only on the transport and transformation of aggregates to understand how their concentration variations and interactions drive the dynamic self-maintenance of the autopoietic individual. Our example makes use of six transformations and two transports (Table 1). The small chains, on the inner side of the membrane, walk at random while remaining attached to the membrane. Because the membrane may wrinkle randomly, to the point where it is possible to remove one of its units while restoring its continuity, these chains gradually grow by taking a membrane unit whenever the membrane is folding next to them (transformation 1). Thus, this chain, initially one particle in length, elongates while remaining attached to the membrane. Once a sufficient size, the chain is transformed into a tetramer that is released into the internal environment (transformation 2). The whole chain eventually disappears in this operation. When several tetramers are side by side, one of them is destroyed and transformed into two dimers (transformation 3). The membrane allows trimers to enter and dimers to exit (transports 1 and 2). In the external environment, a built-in mechanism transforms dimers into trimers. Further, when a trimer and a tetramer are simultaneously present near the membrane, the trimer is converted into a dimer and the membrane takes up the remaining particle; the tetramer is unchanged (transformation 4). When two trimers are simultaneously present near the internal side of the membrane, one of them is converted into a dimer, with the remaining

particle producing a small chain (one-particle length) attached to the inner face of the membrane. The other trimer remains unchanged (transformation 5). When multiple small chain of one particle length are arranged side by side, two of them are moved aside and transformed into a dimer (transformation 6).

### Summary and further analysis

The set of transports, transformations, and movements we have described allows each part of this “cell” (membrane, chains, dimers, trimers, tetramers) to be mobile, linked to others, and constantly destroyed and renewed by others. The whole moves randomly and keeps its size, shape, composition, despite changes in the external environment<sup>5</sup>. The observation that small aggregates move farther than large ones is an indication that this model approximates well Brownian motion. Two individuals simultaneously evolving in the same space maintain their separate identities. Consistent with our definition, this constitutes an autopoietic individual<sup>6</sup>.

Why does this individual retain its size, shape, and composition? To answer this question, we must describe the scheme of regulations guiding our representation, which is, in fact, not specific to the representation above. Indeed, another manifestation hardware or virtual setup (e.g., a cubic matrix using other forms) could be regulated in exactly the same way.

We established several controls for our model. First, synthesis of the membrane is regulated by the concentration of tetramers: the growth of the membrane depends on the presence of a tetramer, but tetramers are produced by the destruction of the membrane. Thus, membrane synthesis cannot occur without previous membrane destruction. The yield from the destruction of tetramers plays a key role in the

<sup>5</sup> As the individual state remains constant, we hypothesize that its global entropy (as a state function) remains unchanged, while that of the environment increases. The representation used here enables their precise calculation, but this has not yet been completed.

<sup>6</sup> A demonstration version and additional documents are available at <https://sites.google.com/site/morphautomaton/>.

regulation of the size of the individual. If the efficiency of this reaction is increased slightly, the number of tetramers decreases. As the membrane synthesis reactions depend on the presence of a tetramer, they become more rare than the reactions of destruction. Accordingly, the size of the membrane decreases. Another effect appears then: as the destruction of the membrane is dependent on the quantity of available membrane, i.e., of its size, this reaction becomes less frequent. A new equilibrium is therefore established when the membrane has reached the size for which these two processes (construction and destruction) become balanced.

Conversely, if we now decrease the efficiency of the reaction of destruction of tetramers, their concentration increases. Therefore, the membrane synthesis reactions that they facilitate become more frequent than the destruction reactions, and the membrane size increases. However, as this size is important, destruction reactions become more frequent. There again, a new equilibrium is established when the membrane has reached the size for which these two processes become balanced.

Additionally, the four processes of synthesis, catabolism, inputs, and outputs are in competition with one another, since the membrane has a finite extent. Several parameters (e.g., the shape of the membrane, the available free space) may favor one process over the others. Finally, many other regulations, unexpected at first sight and unintentionally introduced, directly or indirectly modulate each transformation.

The thermodynamic approach of Virgo and Harvey (2008) proposes a relationship between the amount of energy that can be extracted from the environment and the overall rate at which that energy is used. We searched within our model for such a relationship as a negative correlation between the “activity” of the cell (quantity of metabolic rules executed in a given amount of time) and the potential energy (trimers in the environment) available at the same time, but we did not observe this relationship. However, this first trial must be refined.

This analysis is just beginning. Note, as in physiology, the existence of two kind of regulations whose evolution is either exponential or periodic; the latter probably limits the area where the former can grow.

## Discussion

A schematic representation of autopoiesis was created using a new type of discrete spatial automaton, which is based on the principle of encoding functions through an extensive representation of abstract forms used here as symbols. The identities and localizations of these forms give a complete, static description of the system; the operations performed on them give a dynamic description of the system. While our intent was to represent autopoiesis, this platform is versatile and convenient for representing other biological or non-biological phenomena, especially those observed in any complex population ranging from solutions of macromolecules to swarms or societies.

Despite an as-yet incomplete analysis of this model, it conforms to our definition of an autopoietic individual: each

of its parts is distinct, mobile, linked to the whole, and continuously renewed by one another. In contrast to previous models, the destruction of the membrane, renewal of the catalyst, transmembrane transports, and, therefore, the individual's size are controlled by the individual itself. Additionally, in our model the continuity of the membrane is ensured; the membrane shape, length, and the internal density are effectors and subject to regulations. Indeed, this individual is “robust” since it can adapt to various environmental conditions. In accordance with McMullin's criterion (2004), each individual remains distinct from another one. Thus, we have defined, demonstrated, and described a virtual autopoietic individual. This model represents an important advance in the field, as none of those properties existed in Varela's initial model, where membrane destruction occurred spontaneously and the other parameters were not considered. Further, the original idea of boundary has been replaced by the less-restrictive idea of link. Finally, the concepts discussed here are not restricted solely to biochemistry but could apply to robotics or any other implementation. Presently, this work provides a reference point for the representation of other individuals, their comparisons, their comprehensive “physiological” analysis, the research of a rigorous definition and formalization of autopoiesis, and the potential to undertake the study of the relationship of autopoiesis to other biological properties. This work was developed with a bottom-up approach. The utility of our platform in representing essential interactions of real systems must now be evaluated via a top-down approach.

The purpose of this work was to generate a system, devoid of any emergent property, which could be fully analyzed and controlled. This aim is significant, as clinicians or engineers cannot use devices that may produce some emergent properties beyond their control. Even with an incomplete analysis, we have, indeed, achieved some control in our model. Approaches that aim to produce, starting from scratch, real biological artificial systems potentially able to generate new emergent properties (Rasmussen et al., 2003) or to observe the apparition of emergent structures in virtual systems [i.e., Conway's Game of Life (Beer, 2011) or Swarm Chemistry (Sayama, 2009)] have a different goal: Their aim is to reproduce properties of interest or to explore or generate new ones, but not, at first, to fully explain or control them. Therefore, an analytic phase is necessary before any application of these works can be envisioned.

To consider more fully how this work may be developed and its usefulness as a tool for theoretical analysis, we must place it in context. At the present time, only partial representations of some biological properties are attainable. A complete representation of any living body—even the smallest—with enough definition that the movements of its molecules reproduces its biological properties is out of reach as molecular dynamic simulations require months of computing time to calculate the movements of a small protein during 1 millisecond (Broderick et al., 2005; Shaw et al., 2010). As a consequence, any representations of known biological structures faces the granularity problem: one must decide what real object each bit or tile will represent, creating a potentially endless hesitation between atoms or macro-

structures. Another problem, once this choice has been made, is the lack of compatibility between the representations of several different properties resulting from different analyses, which makes their re-combination impossible to study later (Hucka et al., 2003). The abstract representation proposed here is a consequence of these constraints: it aims to minimize the representation of the structures using a set of abstract forms—similar to symbols—to capture the essentials of the real network interactions.

Notably, four distinct languages were successively constructed in this work (three used here, one remains to be attained). The first deals with the basic representation of space (tiles containing links and associated operations) and offers only the possibility of combining them to represent more elaborate forms. The second language deals with this potentially limitless set of complex forms (made of combinations of tiles and links) and the operations performed on them (identification, movement, transformation, transport). The third language deals with the schematic representation of the system. The only information the symbols in this language should carry is their identity and localisation. The operations describe their movements and possible transports and transformations. The fourth should describe, in their simplest forms, the laws common to any similar system (here, all autopoietic individuals sharing the same set of regulations). As there exists the possibility to combine several representations developed within the same parameters, our platform could be convenient to individually represent distinct properties of biological systems as well as combinations of them.

**Acknowledgements.** The author would like to thank Prof. Michel Morange for elaborate discussions and helpful advice, Dr. Barry McMullin for his useful critique on the conceptual aspects, Dr. René Doursat, Gilbert Lechermeier, Christophe Malaterre, Sarah Maurer, Jacob Spangler, and Dr. Jean-Marc Victor for their valuable comments, Sebastien Boisard who wrote the first rules parser (1994), and Dr. Sheila Cherry (Fresh Eyes Editing) who provided a constant professional support and managed to make this collaboration enjoyable.

## References.

Atlan, H. (2011). *Le vivant post-genomique ou qu'est-ce que l'auto-organisation ?*, Odile Jacob Sciences.

Beer, R. D. (2011). Autopoiesis and cognition in the game of life. *Artif Life* **10**(3): 309-326.

Bersini, H. (2010). Software replica of minimal living processes. *Orig Life Evol Biosph* **40**(2): 121-130.

Broderick, G., M. Ru'aini, et al. (2005). A life-like virtual cell membrane using discrete automata. *In Silico Biol* **5**(2): 163-178.

Goodsell, D. S. (2009). *The machinery of life*. New York, Copernicus Books.

Hucka M., A. Finney, et al. (2003). The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* **19**(4): 524-31.

Ikegami, T. and K. Suzuki (2008). From a homeostatic to a homeodynamic self. *Biosystems* **91**(2): 388-400.

Kleiber, M. (1947). Body size and metabolic rate. *Physiological Reviews* **27**(4): 511-541.

Liu Y. Y., J. J. Slotine, and A. L. Barabasi (2011). Controllability of complex networks. *Nature* **473**: 167-73.

Maturana, H. R. and F. Varela (1973). *Autopoiesis: the organization of living systems*. Dordrecht, Holland, D. Reidel Publishing Company.

Maturana, H. and F. Varela (1980). *Autopoiesis and cognition: the realization of the living*. D. Reidel Publishing Co, Dordrecht.

McMullin, B. (2004). Thirty years of computational autopoiesis: a review. *Artif Life* **10**(3): 277-295.

McMullin, B. and D. Gross (2001). *Towards the implementation of evolving autopoietic artificial agents*. 6th European Conference on Artificial Life Prague.

McMullin, B. and F. Varela (1997). *Rediscovering computational autopoiesis*. 4th European Conference on Artificial Life, Brighton, UK, MIT press.

Morange, M. (2005). *La vie expliquée? 50 ans apres le double helice*, Odile Jacob Sciences.

Morris, M. K., J. Saez-Rodriguez, et al. (2010). Logic-based models for the analysis of cell signaling networks. *Biochemistry* **49**(15): 3216-3224.

Popa, R. (2004). *Between necessity and probability: searching for the definition of life*. New York, Springer-Verlag.

Rao, C. V., D. M. Wolf, et al. (2002). Control, exploitation and tolerance of intracellular noise. *Nature* **420**(6912): 231-237.

Rasmussen, S., L. Chen, M. Nilsson, and S. Abe (2003). Bridging living and nonliving matter. *Artif Life* **9**: 269-316.

Sayama, H. (2009). Swarm chemistry. *Artif Life* **15**: 105-114.

Schoenheimer, R. (1942). *The fire of life: the dynamic state of body constituents*. Cambridge, MA, Harvard University Press.

Schrödinger, E. (1944). *What is life? The physical aspect of the living cell*. Cambridge [Eng.], The University press.

Shaw, D. E., P. Maragakis, et al. (2010). Atomic-level characterization of the structural dynamics of proteins. *Science* **330**(6002): 341-346.

Virgo, N. and I. Harvey. (2008). Reaction-diffusion spots as a model for autopoiesis. *Artif Life* **11**: 816.

Wishart, D. S., R. Yang, et al. (2005). Dynamic cellular automata: an alternative approach to cellular simulation. *In Silico Biol* **5**(2): 139-161.

Zeleny, M. (1977). Self-organization of living systems: A formal model of autopoiesis. *International Journal of General Systems* **4**: 13-28.

# Ouroboros avatars:

## A mathematical exploration of Self-reference and Metabolic Closure

Jorge Soto-Andrade<sup>1</sup>, Sebastián Jaramillo<sup>2</sup>, Claudio Gutiérrez<sup>3</sup> and Juan-Carlos Letelier<sup>2</sup>

<sup>1</sup> Departamento de Matemáticas, Facultad de Ciencias, Universidad de Chile, Casilla 653, Santiago, Chile

<sup>2</sup> Departamento de Biología, Facultad de Ciencias, Universidad de Chile, Casilla 653, Santiago, Chile

<sup>3</sup> Departamento de Ciencias de la Computación, Universidad de Chile, Blanco Encalada 2120, Santiago, Chile  
sotoandrade@u.uchile.cl

### Abstract

One of the most important characteristics observed in metabolic networks is that they produce themselves. This intuition, already advanced by the theories of Autopoiesis and  $(M,R)$ -systems, can be mathematically framed in a weird looking equation, full of implications and potentialities:  $f(f) = f$ . This equation (here referred as *Ouroboros equation*), arises in apparently dissimilar contexts, like Robert Rosen's synthetic view of metabolism, hyperset theory and, importantly, untyped lambda calculus. In this paper we survey how Ouroboros equation appeared in those contexts, with emphasis on Rosen's  $(M,R)$ -systems and Dana Scott's work on reflexive domains, and explore different approaches to construct solutions to it. We envision that the ideas behind this equation, a unique kind of mathematical concept, initially found in biology, would play an important role towards the development of a true systemic theoretical biology.

### Introduction

Ouroboros (also written Uroboros), the ancient symbol of the snake eating its own tail, is often taken nowadays to represent self-reference and circularity. In this vein we call in this paper "Ouroboros equation", the ultimate self-referential equation  $f(f) = f$ .

Notice that  $f$  (supposedly a function) applies to itself, as an argument, the result being again  $f$ . So  $f$  plays simultaneously the roles of argument, function and value.

Recall that equation solving in mathematics has a long history, beginning with equations like  $2x = 1$ ,  $x + 3 = 1$ , up to  $x^2 = 2$  and  $x^2 = -1$ .

Each of these equations was solved introducing new species of numbers, some of them meeting strong resistance, like negative and imaginary numbers. Indeed methods developed to construct the irrational  $\sqrt{2}$  and the imaginary  $\sqrt{-1}$  may serve as metaphors to tackle the bigger and subtler challenge of constructing somehow solutions of Ouroboros equation  $x(x) = x$ . Since this equation suggests that  $x$  should be some sort of function, we will write it

$$f(f) = f.$$

in the sequel. However the main motivation to consider Ouroboros equation did not arise from everyday mathematics proper. It arose from various fields ranging from

Logic and Computer Science to Theoretical Biology. For these reasons, we call "Ouroboros avatars", the various manifestations or ways in which Ouroboros equation has emerged in different domains (although "avatar" means in fact "descent" in Sanskrit). We have then avatars of Ouroboros in Logic (Löfgren, 1968; Scott, 1972, 1973), Hyperset Theory (Aczel, 1988), Cognitive Sciences (Kampis, 1995; Kauffman, 1987), Computer Science and Informatics (Scott, 1972; Kampis, 1995; Milner, 2006), Systems Theory and Theoretical Biology (Rosen, 1991; Soto-Andrade and Varela, 1984; Maturana and Varela, 1980; Letelier et al., 2006, 2005), and others, that we review in the next sections.

A most remarkable fact, commented below, is the similarity of methods of constructing solutions to Ouroboros, developed in fields apparently as unrelated as logic (Scott, 1972, 1973) and metabolic systems theory (Letelier et al., 2006, 2005), motivated by the construction of actual mathematical models for untyped lambda calculus and virtual infinite regress in metabolic systems, respectively.

### Ouroboros is not an oxymoron

To begin with, it can be proved that Ouroboros is not an oxymoron, i.e. that the existence of an object  $f$  such that  $f(f) = f$ , belonging to its own domain and range, is not logically inconsistent (Löfgren, 1968; Kampis, 1995). It had been argued nevertheless that this was impossible (Wittgenstein, 1961) or paradoxical (Rosen, 1959). Instead, it turns out that an atomically self-reproducing entity can be axiomatized, and in this sense it really does exist (Löfgren, 1968). In fact Löfgren (1968) has shown that the axiom of complete self-reference is independent from usual set theory and logic, and can therefore be added to it as a new primitive axiom, that it is impossible to derive from the other axioms. Solutions to Ouroboros, as Quine's atoms  $Q = \{Q\}$  (Quine, 1980), appear then as completely self-referential, inapproachable, a perfectly closed class in itself (Kampis, 1995). Varela takes a similar stance, when he introduces self-referentiality from scratch as a third mark for self-indication or autonomous value (Varela, 1975), extending the indicational calculus of Spencer Brown (1969), and

later as a third logical value, besides true and false (Varela, 1979; Kampis, 1995).

Our viewpoint is however that Ouroboros lives indeed outdoors, with respect of our usual logical - mathematical realm, but just outside, in front of the door, say, so that it can be *approximated* stepwise "from within". This intuition has been captured to a great extent, in different guises, in Scott (1972, 1973); Soto-Andrade and Varela (1984), in Varela's further work (Varela and Goguen, 1978) and in Letelier et al. (2006), as we explain below.

## Ouroboros in Self-referential formalisms

As already said, Ouroboros equation  $f(f) = f$  involves self-reference, or more precisely, recursion (for a systematic overview of fields that deal with different forms of self-reference see Kauffman (1987)).

An interesting notion of recursion arises when dealing with its operative issues. This approach, linked with the theory of computing, has a strong relationship with the notion of *application*. It is not surprising that formalisms for abstracting the notions of function and program, like lambda calculus and the theory of recursion, are at the center of these developments.

The paradigmatic theory of functional application is the *simple* lambda calculus (i.e. with no distinction of types) (Barendret, 1984), introduced by Church (1951). In the untyped lambda calculus the equation  $f(f) = f$  has a trivial solution:  $\lambda x.x$ , that is, the identity function. The crucial point here is the absence of typing, something that cannot be realized with the identity function in classical mathematical structures (like vector spaces, groups, etc.), where argument and function belong to different types.

The very essence of the power of this formalism resides in that it overcomes the traditional mathematical notion of function as a set of pairs (input, output), by focusing instead on the composition and evaluation of functions. So formalisms like the lambda calculus are much better suited for the formalization of fields where the process of evaluation is most relevant or even the core of the phenomenon itself. Lambda calculus was disregarded by the logical and mathematical communities until the seventies. What brought their attention to lambda calculus was the work of Dana Scott providing mathematical models for this formalism. The idea is simple (not so much its implementation however...): finding spaces where these objects (lambda terms, that is, generalized functions) may live. To see the difficulties, let us exemplify the hierarchy of objects that can be created from a set  $U$ : functions with zero parameters (these are the elements of  $U$ ); functions with one parameter, that is,  $f : U \rightarrow U$ ; functions with two parameters,  $g : U \times U \rightarrow U$  and so on. All of them can be expressed in lambda calculus, that is, they should be elements of the wanted space  $D$ . In particular, in this typeless environment it should be possible to apply a function  $f : D \rightarrow D$  to itself, as another element of  $D$ .

It is worth reviewing the basic construction in Scott (1972, 1973), where continuity and limits play a central role, by restricting the universe of functions to be considered. The central question is:

"Are there nontrivial spaces  $D$  that can be identified (as topological spaces) with their function spaces  $[D \rightarrow D]$ , consisting of all continuous functions from  $D$  to  $D$ ?"

Scott showed that indeed there are many of them, and called them "reflexive domains". His idea was to start with a space  $D_0$ , with suitable properties (e.g. a continuous lattice), and try to identify its function space  $D_1 = [D_0 \rightarrow D_0]$  with  $D_0$ . A difficult task indeed, but we may notice that  $D_0$  can be embedded in  $D_1$ , by identifying each element  $d_0 \in D_0$  with the constant function in  $D_1$  with value  $d_0$ , and also that  $D_1$  can be projected onto  $D_0$  by sending each (continuous) function  $d_1 \in D_1$  to its minimum value  $d_1(\perp)$  (where  $\perp$  is the least element of the complete lattice  $D_0$ ). Call  $i_0$  and  $p_0$  the embedding and the projection so defined. This allows us to embed in a clever way  $D_1 = [D_0 \rightarrow D_0]$  into  $D_2 = [D_1 \rightarrow D_1]$ , by sending each  $d_1$  to  $i_0 \circ d_1 \circ p_0$  and - dually - to project  $D_2$  onto  $D_1$  by sending  $d_2$  to  $p_0 \circ d_2 \circ i_0$  and so on, to obtain iteratively a double chain of embeddings from  $D_n$  into  $D_{n+1} = [D_n \rightarrow D_n]$ , and projections from  $D_{n+1}$  onto  $D_n$ , for all  $n$ . We obtain then the wanted reflexive domain as the limiting space  $D_\infty$  of this double sequence of continuous maps between continuous lattices.

Regarding our interest here, the later result shows that there is a space where Ouroboros equation at least makes sense, i.e. it "types". To the best of our knowledge, Scott did not consider this equation explicitly, although several notions of his come close to it.<sup>1</sup>

Scott's construction inspired the limiting construction of a self-referential extension of Spencer Brown (1969) calculus of indications by Varela and Goguen (1978), where they endow the collection of all forms that can be constructed in Brown's setting with the same sort of structure that Scott (1972) considered, i.e. chain complete partially ordered sets (posets). In their setting fully self-referential equations like Ouroboros' would have solutions. That is a different way to extend Brown's setting that the one in Varela (1975).

Scott's construction also inspired later the construction of reflexive domains in the context of posets and monotone mappings with suitable continuity properties, carried out in Soto-Andrade and Varela (1984), where the relationship between the existence of fixed points and several instances of self - reference is also discussed (notice that a reflexive domain  $D$  is a fixed point for the function  $D \mapsto [D \rightarrow D]$ ).

Another formalism where Ouroboros equation arises naturally is hyperset theory (also called non well founded set theory). Hypersets constitute an extension of usual set theory, that allow sets to be members of themselves, like Quine's atom  $Q = \{Q\}$  (Quine, 1980; Aczel, 1988). We

<sup>1</sup>See for example Proposition 3.14 in Scott (1972)

meet among them baby Ouroboros like  $f = \{(f, f)\} = \{\{f, \{f\}\}\}$ , that satisfy  $f(f) = f$ , if we identify the function  $f$  with its graph and choose the usual set theoretical model  $\{a, \{b\}\}$  for ordered pairs  $(a, b)$ .

As discussed in Löfgren (1968) and Kampis (1995), self-reference is closely tied to language. Hence it is not surprising that formalisms that allow to break the classical hierarchies between language and metalanguage, or as in hyper-sets, between container and containee, can provide solutions to the Ouroboros equation. Up to now however, these formalisms do not seem to have been meaningfully exploited in the context of biological self-reference and circularity (see Cárdenas et al. (2010) more a more detailed survey).

### Ouroboros in $(M, R)$ systems: Infinite regress face to face

We turn now to Rosen's synthetic insights regarding metabolic circularity, that he developed completely independently of Scott (for a comprehensive survey of references about Rosen's work see Cárdenas et al. (2010)). In his formalism of  $(M, R)$  systems, the collective action of the thousands of catalysts in a metabolic network  $M$  coalesces into a single mapping  $f$  from  $A$ , the collection of all sets of reactants, to  $B$ , the collection of all sets of products, that transforms inputs  $a \in A$  into outputs  $b = f(a) \in B$ .

But in any metabolic system, catalysts are subject to degradation, wear and tear, and therefore need to be regenerated or replaced by the system. To meet this requirement, Rosen looked upon the replacement mechanism as a procedure, denoted by  $\Phi$ , that, from a suitable  $b = f(a) \in B$  as input, reproduces  $f$  according to  $\Phi(b) = f$ . Because the net effect of  $\Phi$  is to select from the relatively large set  $H(A, B) \subset \text{Map}(A, B)$ , of all possible metabolisms, a specific  $f$  such that  $f(a) = b$ , using  $b \in B$  as an input, Rosen calls it a *selector*. Thus, the procedure  $\Phi$  representing replacement appears as a map from  $B$  to  $H(A, B)$ .

Then an  $(M, R)$  system has the following algebraic description based on two mappings  $f, \Phi$  acting in synergy:

$$\begin{aligned} A &\xrightarrow{f} B \xrightarrow{\Phi} H(A, B) \\ a &\mapsto f(a) = b \mapsto \Phi(b) = f \end{aligned}$$

But now, it is possible to go further and demand the system to be capable of replacing the replacer, or selector,  $\Phi$ : a replicative  $(M, R)$  system in Rosen's terminology (this property is also referred as *organizational invariance* (Cárdenas et al., 2010)). More precisely,  $\Phi$  should be generated with the help of a procedure that, given a metabolism  $f$ , produces the corresponding  $\Phi$  that selects metabolism  $f$ , that is a mapping  $\beta : H(A, B) \rightarrow H(B, H(A, B))$  such that  $\beta(f) = \Phi$ , and so on... The big question is then, how can this be, without implying infinite regress?

Rosen's solution to avoid infinite regress, was to posit that the equation  $\Phi(b) = f$  is to have only one solution  $\Phi$  (a

most demanding constraint indeed!) so that the mapping  $\beta$  sends  $f$  to this unique selector  $\Phi$ . In other words,  $\beta$  is "just" the inverse of the "evaluation at  $b$ " operator (acting on functions whose domain contains  $b$ ) so that no further procedure is needed to construct  $\beta$  itself. It is in this sense that Rosen claims that his construction solves the problem of infinite regress. Rosen was however unable to give concrete examples where this hypothesis was fulfilled.

The operation of an *organizationally invariant*  $(M, R)$  system can therefore be viewed as three mappings  $(f, \Phi, \beta)$  acting in synergy:

$$\begin{aligned} A &\xrightarrow{f} B \xrightarrow{\Phi} H(A, B) \xrightarrow{\beta} H(B, H(A, B)) \\ f(a) &= b, \quad \Phi(b) = f, \quad \beta(f) = \Phi. \end{aligned}$$

where  $\beta$  is the inverse of the "evaluation at  $b$ " operator.

Now, if instead of shunning infinite regress, as Rosen did, we look at it "face to face", a recursive construction emerges, whose first step is motivated by the question:

If you have a map  $f : A \rightarrow B$ , can you find a new map  $f_1 : B \rightarrow C$  such that for a suitable  $a \in A$  you have  $f_1(f(a)) = f$  or, equivalently  $f_1(b) = f$ ;  $b = f(a)$ ?

Of course, the answer to this question, taken at face value, when  $A, B$  and  $C$  are plain (unstructured) sets and  $f$  and  $f_1$  are set mappings, is "Obviously, yes", since you have plenty of maps from one set to another which take a prescribed value on a given point. Just take  $C$  to be the set  $\text{Map}(A, B)$  of all mappings from  $A$  to  $B$  and  $f_1$  to be any mapping from  $B$  to  $C$  such that  $f_1(b) = f$ .

However this question becomes more intelligent when stated in a categorical framework, typically when we consider our sets endowed with some sort of structure and have our maps preserve this structure.

Then, if we take our structured sets to be vector spaces, our maps would be linear; if our sets are posets (i.e. partially ordered sets), our maps ought to be monotone (order preserving). If our sets were endowed with a metric, or distance, then our allowed mappings might be continuous, or even "isometric", i. e. "distance - preserving" mappings. Structure preserving mappings are usually called "homomorphisms". For instance, the homomorphisms between vector spaces are linear mappings.

Now we can state the categorical version of our question: In a category (of structured sets and structure preserving mappings, say), given a homomorphism  $f : A \rightarrow B$ , can you find a new homomorphism  $f_1 : B \rightarrow C$  such that for a suitable  $a \in A$  you have  $f_1(b) = f$ , where  $b = f(a)$ ?

The subtlety now lies in the fact that to carry over our obvious set theoretical solution to the categorical setting, we need to find among all mappings  $f_1$  such that  $f_1(b) = f$ , one which is well behaved enough to be a *homomorphism* from the structured set  $B$  to *another structured set*  $C$ . We would be happy then to know that the set  $H(A, B)$ , consisting of all homomorphisms from  $A$  to  $B$ , may be endowed with the same (type of) structure than  $A$  and  $B$ . If it is the case, we would take  $C$  to be  $H(A, B)$ , and we would be

all set up to seek a homomorphism  $f_1$  from  $B$  to  $C = H(A, B)$ , which takes the value  $f$  at point  $b \in B$ .

Recall now that Rosen, to avoid infinite regress, posited the uniqueness of such a function  $f_1$ , called  $\Phi$  in his setup (Rosen, 1991; Letelier et al., 2006).

It is clear however that in the category of sets, where the existence of such an  $f_1$  is obvious, uniqueness is impossible (unless  $B$  is a singleton). Nevertheless, if you change the underlying category (i.e. the stage for the problem) so as to have a category whose sets of homomorphisms  $H(X, Y)$  are much smaller than  $Map(X, Y)$ , i.e. become more and more selective, existence may become less and less obvious and uniqueness may become more and more possible.

We may hope then for the existence of a *turning point* in the choice of our category, at which the sets of homomorphisms  $H(X, Y)$  would have the right size so as to have simultaneously existence and uniqueness of our homomorphism  $f_1$ . Rosen's dream was that such *turning points* (or better, *turning categories*) exist, where his hypothesis would be fulfilled! They might indeed be dubbed "metabolic categories."

If we look however infinite regress face to face and we do not care about uniqueness, we could continue our construction above forever, in the spirit of Soto-Andrade and Varela (1984) under a mild hypothesis of existence of our homomorphisms  $f_1$ , in the framework of a *concrete category*  $\mathcal{C}$ , i.e. a category of structured sets and structure preserving maps (the only ones that we will consider in this article).

### Hypothesis 1. (Existence of "replacing homomorphisms")

We assume that given any homomorphism  $f : A \rightarrow B$  in our concrete category  $\mathcal{C}$ , we can choose  $a \in A$  such that the following hold:

- there exists a homomorphism  $f_1 : B \rightarrow H(A, B)$ , such that  $f_1(f(a)) = f$  (we say then that  $a \in A$  is an  $f$ -generic element),
- there exists a homomorphism  $f_2 : H(A, B) \rightarrow H(B, H(A, B))$ , such that  $f_2(f_1(f(a))) = f_1$  (i.e.  $f(a)$  is  $f_1$ -generic), and so on...

Notice that this hypothesis requires implicitly that,  $A$  and  $B$  being any objects in  $\mathcal{C}$ , the set of homomorphisms  $H(A, B)$  should also be an object in  $\mathcal{C}$ , i.e. it can be endowed with the same structure as  $A$  and  $B$ . Also, simple examples (see below) show that it is not to be expected that every  $a \in A$  be  $f$ -generic for a given  $f : A \rightarrow B$ .

**Example 1.** In the category of (finite dimensional) vector spaces and linear mappings, our hypothesis is clearly fulfilled. Indeed, if  $f$  is the null mapping 0, we just take  $a = 0$  and  $f_1, f_2, \dots$  to be 0 all the way. If  $f \neq 0$ , take  $a$  to be any non zero vector in  $A$ , such that  $f(a) \neq 0$  and then  $f_1$  to be any linear mapping from  $B$  to  $H(A, B)$  sending  $f(a)$  to  $f$ ,  $f_2$  to be any linear mapping sending  $f$  to  $f_1$ , and so on. These (non zero!) linear maps exist recursively by the well known elementary "linear extension property" for finite

dimensional vector spaces, saying that you can always construct linear mappings from one vector space  $V$  to another that take a prescribed value at a given non zero vector in  $V$ .

**Example 2.** In the category of additive groups and addition preserving maps, we take  $A = B = \mathbb{Z}_3^+$ , the set of integers  $0, 1, 2 \bmod 3$  endowed with the operation  $+$  of addition mod 3. Notice that  $1 + 1 + 1 = 0 \bmod 3$ . Then  $H(A, A) = \{h_a | a \in A\} \simeq A$ , where  $h_a$  is the "scaling map" with ratio  $a$ , that sends  $b$  to  $ab$  ( $b \in A$ ), which we identify with  $a \in A$ , writing  $h_a = a$ . So we identify the mapping  $h_a$  with its value  $a$  at 1. The set  $H(A, A)$  endowed with the operation of addition of mappings is also an additive group, isomorphic to  $A$ , and  $h_a + h_b = h_{a+b}$  ( $a, b \in A$ ).

If we take now  $f$  to be the null mapping  $h_0 = 0$ , we see that for any  $a \in A$ , every  $f_1 : A \rightarrow H(A, A)$ , satisfies  $f_1(f(a)) = f$ , since  $f_1(f(a)) = f_1(0) = 0 = h_0 = f$ . Hence any  $a \in A$  is  $h_0$ -generic and we may take  $f_1$  to be  $h_0, h_1$  or  $h_2$  (i.e. such that  $f_1(1) = h_0, h_1$  or  $h_2$ ). The choice of  $f_1$  becomes relevant when we go one step further, asking now for a homomorphism  $f_2 : H(A, A) \rightarrow H(A, H(A, A))$  such that  $f_2(f) = f_1$ . In a diagram:

$$\begin{array}{ccccc} A & \xrightarrow{f} & A & \xrightarrow{f_1} & H(A, A) & \xrightarrow{f_2} & H(A, H(A, A)) \\ a \mapsto f(a) & \mapsto & f & \mapsto & f_1 \end{array}$$

Indeed, since  $f = h_0$ , we have that necessarily  $f_2(f) = f(0) = 0 = h_0$ , so  $f$  is  $f_1$ -generic only for  $f_1 = h_0$ , but not for  $h_1$  or  $h_2$ . On the other hand, if we begin with  $f = h_2$  instead of  $h_0$ , then for any non zero  $a \in A$ , we find a *unique*  $f_1 : A \rightarrow H(A, A)$  such that  $f_1(f(a)) = f$ , since the equation amounts to  $f_1(2a) = 2$ , i.e.  $x2a = 2$ , i.e.  $x = a^{-1}$ , if we write  $f_1 = h_x$ . So every non zero  $a \in A$  is  $f$ -generic in this case but 0 is not, since  $f_1(f(0)) = h_0$ .

Applying now our hypothesis recursively, we can construct the following infinite sequence of homomorphisms (and objects) in our concrete category  $\mathcal{C}$ , issued from any homomorphism  $C_0 \xrightarrow{\Phi_0} C_1$  in  $\mathcal{C}$ :

$$\begin{array}{ccccccccccc} C_0 & \xrightarrow{\Phi_0} & C_1 & \xrightarrow{\Phi_1} & C_2 & \xrightarrow{\Phi_2} & C_3 & \xrightarrow{\Phi_3} & \dots & \xrightarrow{\Phi_{n-1}} & C_n & \xrightarrow{\Phi_n} & C_{n+1} & \xrightarrow{\Phi_{n+1}} & \dots \\ c_0 & \rightarrow & c_1 & \rightarrow & c_2 & \rightarrow & c_3 & \rightarrow & \dots & \rightarrow & c_n & \rightarrow & c_{n+1} & \rightarrow & \dots \end{array}$$

satisfying the following:

$$C_2 = H(C_0, C_1), \dots, C_{n+1} = H(C_{n-1}, C_n)$$

so that  $\Phi_n \in H(C_n, C_{n+1}) = C_{n+2}$ ,

$$\Phi_1(\Phi_0(c_0)) = \Phi_0 \quad \text{for a suitable } c_0 \in C_0,$$

$$\Phi_n(c_n) = c_{n+1} \in C_{n+1} \quad (n \geq 0) \quad \text{and}$$

$$\Phi_{n+1}(\Phi_n(c_n)) = \Phi_n \quad \text{for all } n \geq 1;$$

Notice that to have consistent notations, we have renamed  $A$  to  $C_0, B$  to  $C_1, C$  to  $C_2; f$  to  $\Phi_0, f_1$  to  $\Phi_1$ .

Moreover, since  $\Phi_0(c_0) = c_1$  we have  $\Phi_0 = \Phi_1(c_1) = c_2$ , and inductively,

$$\Phi_n = \Phi_{n+1}(\Phi_n(c_n)) = \Phi_{n+1}(c_{n+1}) = c_{n+2} \quad (n \geq 0),$$

in other words,  $c_n = \Phi_{n-2}$  for all  $n \geq 2$ , so that

$$\Phi_{n+1}(\Phi_n(c_n)) = \Phi_{n+1}(c_{n+1}) = \Phi_{n+1}(\Phi_{n-1}) = \Phi_n,$$

showing how the homomorphisms  $\Phi_n$  play here alternatively the role of argument, function and value...



We have then three different but equivalent ways to state the recursive relationship between the  $\Phi_n$ 's:

1.  $\Phi_{n+1}(\Phi_n(c_n)) = \Phi_n$
2.  $\Phi_{n+1}(\Phi_{n-1}) = \Phi_n$
3.  $\Phi_{n+1}(c_{n+1}) = \Phi_n$

Remark now that the last one may be written

$$ev_{c_{n+1}}(\Phi_{n+1}) = \Phi_n$$

in terms of the “evaluation at  $x$ ” mappings  $ev_x : f \mapsto f(x)$ . So the following “reverse” sequence of mappings and elements emerges, where each  $C_n$  “projects” onto  $C_{n-1}$ :

$$\begin{array}{ccccccc} C_1 & \xleftarrow{ev_{c_0}} & C_2 & \xleftarrow{ev_{c_1}} & C_3 & \xleftarrow{ev_{c_2}} & \dots \xleftarrow{ev_{c_{n-2}}} C_n \xleftarrow{ev_{c_{n-1}}} \dots \\ c_1 & \xleftarrow{ev_{c_0}} & \Phi_0 & \xleftarrow{ev_{c_1}} & \Phi_1 & \xleftarrow{ev_{c_2}} & \dots \xleftarrow{ev_{c_{n-2}}} \Phi_{n-2} \xleftarrow{ev_{c_{n-1}}} \dots \end{array}$$

This sequence of evaluation maps  $ev_{c_n}$  forms what mathematicians call a *projective (or inverse) system of mappings*. In the category of sets and mappings, every such system of mappings, call it

$$C_1 \xleftarrow{p_1} C_2 \xleftarrow{p_2} C_3 \xleftarrow{p_3} \dots \xleftarrow{p_{n-1}} C_n \xleftarrow{p_n} C_{n+1} \xleftarrow{p_{n+1}} \dots$$

has a (projective) “limit”, which is rigorously characterized as the set  $C^\infty$  consisting of all sequences  $(c_1, c_2, \dots, c_n, \dots)$  of “coherent” choices of elements  $c_n \in C_n$  (“coherent” meaning here that each  $c_n$  “projects” onto  $c_{n-1}$ , i.e.  $p_{n-1}(c_n) = c_{n-1}$ ). This projective limit set  $C^\infty$  “projects” also in a natural way onto each  $C_n$ , sending each sequence to its  $n$ -th term  $c_n$ . Intuitively, this construction allows us to get hold as elements in the limit set  $C^\infty$ , of “mythical” or “ideal” objects” that cast a series of approximating down to earth “shadows” (the  $c_n$ 's). In concrete categories we may expect moreover that the structure we have on all  $C_n$ 's will carry over to the limit set  $C_\infty$ , which will become then a bona fide object in our category, projecting itself by homomorphisms onto each  $C_n$ .

**Digression: A baby projective limit.** To convey a better insight into projective limits, we recall here a baby example from Soto-Andrade and Varela (1984), that highlights their elementary set theoretical nature.

Consider the increasing nested sequence of finite sets

$$C_n = \{1, 2, \dots, n\} \quad (n = 1, 2, 3, \dots),$$

whose union is the set  $\mathbb{N}$  of all natural numbers. This sequence of sets becomes a projective system if we “project downwards”, or “contract inwards” each  $C_{n+1}$  onto the smaller  $C_n$  by sending every  $m \leq n$  to itself and  $n+1$  to  $n$ . Call these projections (or contractions)  $p_n$ . So on  $C_{n+1}$  we have  $p_n(n) = n = p_n(n+1)$ . The projective limit  $C^\infty$  can be intuited now as the set of all numbers in  $\mathbb{N}$  plus an extra “mythical boundary point”  $+\infty$ , situated at the far right of all natural numbers.

Indeed, going back to the precise definition of  $C^\infty$ , we see that the points  $m \in \mathbb{N}$  appear as “limits” of the sequences of coherent choices  $(1, 2, \dots, m-1, m, \dots, m, \dots)$  that after a while “stutter” indefinitely or become “constant”. But we also have the coherent chain of choices given by  $1 \in C_1$ ,  $2 \in C_2$ ,  $3 \in C_3$ , and so on. Notice that each  $m \in C_n$  is the “ancestor” of the preceding  $m-1 \in C_{m-1}$ .

This sequence of choices represents then our “mythical far right boundary point”  $+\infty$ , whose  $n$ -th projection is  $n$ . Analogously, we may obtain  $\{-\infty\} \cup \mathbb{Z} \cup \{+\infty\}$  as a projective limit. This shows concretely how the projective limit allows us to get hold of “mythical” or “ideal” objects that cast a series of approximating down to earth “shadows”.

Recall that also fractals, a paradigmatic example of “mythical shapes”, may be looked upon in this way, as projective limits of everyday shapes (*loc. cit.*).

**Properties of the limit objects  $C^\infty$  and  $\Phi_\infty$ .** The coherent sequence  $\Phi_n$  in the system of evaluation maps  $ev_{c_n}$  is an element of the projective limit  $C^\infty$ . We call it  $\Phi_\infty$  and we write  $\Phi_\infty = \lim_{n \rightarrow \infty} \Phi_n$  to convey the intuition that  $\Phi_\infty$  is a kind of “limit” of the  $\Phi_n$ 's as  $n$  tends to  $\infty$ . Notice that this quite analogous to the way in which a “rational” person constructs  $\sqrt{2}$  with the help of Cauchy sequences of rational numbers. Now, intuitively, by passing to the limit as  $n$  tends to  $\infty$  in the recursive relation  $\Phi_{n+1}(\Phi_{n-1}) = \Phi_n$  we obtain the stunning self referential equation

$$\Phi_\infty(\Phi_\infty) = \Phi_\infty,$$

saying that  $\Phi_\infty$  is a *solution to Ouroboros equation!*

Analogously, making  $n$  tend to  $\infty$  in the equation  $C_{n+1} = H(C_{n-1}, C_n)$ , we get

$$C_\infty = H(C_\infty, C_\infty),$$

so that  $C_\infty$  is a reflexive domain, as in Soto-Andrade and Varela (1984). We will not go here into the rigorous justification of this passage to the limit, since it involves a more precise description of  $\Phi_\infty$  as a mapping in  $H(C_\infty, C_\infty)$ , taking into account the double system of mappings  $\Phi_n : C_n \rightarrow C_{n+1}$  and  $ev_{c_n} : C_n \leftarrow C_{n+1}$ , as in Scott (1972).

Apparently no mathematician imagined this recursive procedure to construct solutions of Ouroboros equation before Rosen introduced his  $A \xrightarrow{f} B \xrightarrow{\Phi} H(A, B)$  setup as a formal description of metabolism (Rosen, 1958; Letelier et al., 2005). Notice that this construction is quite different although formally analogous to Scott's (Scott, 1972, 1973).

**An arithmetical avatar of Ouroboros.** Generalizing example 2 above, we put  $C_0 = C_1 = A = \mathbb{Z}_m^+$ , the set of integers  $0, 1, 2, \dots, m-1 \bmod m$ , endowed with the operation  $+$  of addition mod  $m$ . Then  $C_2 = H(A, A) = \{h_a | a \in A\} \simeq A$ , where  $h_a : b \mapsto ab$  for all  $b \in A$  and we identify as before each  $h_a$  with  $a$ . We endow  $H(A, A)$  with the operation of addition of mappings.

Now, since recursively  $H(A, A) \simeq A$ ,

$$H(A, H(A, A)) \simeq H(A, A) \simeq A,$$

$$H(H(A, A), H(A, H(A, A))) \simeq H(A, A) \simeq A$$

and so on, we have that all  $C_n$  are isomorphic to  $A$ .

To identify the mappings  $\Phi_n$  we need then only to solve multiplicative equations  $ax = b \bmod m$  in  $A$ . If  $m = 3$ , as in example 2 we choose  $c_0 = 1 \bmod 3$  and  $\Phi_0 = h_2 = 2$ . Then  $c_1 = 2$  and  $\Phi_1 = h_1 = 1$ , and our coherent sequence begins  $1 \xleftarrow{h_2} 2 \xleftarrow{h_1} 2$ . Next, we must look for  $\Phi_2$  such that

$\Phi_2(2) = h_1$ , i.e. for  $a \in A$  such that  $a \cdot 2 = 1$ , so  $a = 2$ .

It follows recursively that our sequence will look like

$$1 \xrightarrow{h_2} 2 \xleftarrow{h_1} 2 \xrightarrow{h_2} 1 \xleftarrow{h_2} 2 \xrightarrow{h_1} 2 \xleftarrow{h_2} 1 \xrightarrow{h_2} 2 \xleftarrow{h_1} 2 \xrightarrow{h_2} \dots$$

so, intuitively,  $\Phi_\infty$  is the “limit” of this “wave like” oscillating sequence, although formally  $\Phi_\infty$  is this sequence.

Notice also that our sequence  $\Phi_\infty$  is a multiplicative analogue mod 3 of the ubiquitous Fibonacci sequence: Instead of  $c_{n+1} = c_n + c_{n-1}$  we have  $c_{n+1} = c_n \cdot c_{n-1} \bmod 3$ .

If we take now  $m = 10$ , for instance, and we put  $c_0 = 3$  and  $\Phi_0 = h_9$ , so that  $c_1 = 7$ , we find recursively that  $\Phi_\infty$  is embodied in the projective sequence

$$3 \xleftarrow{h_9} 7 \xleftarrow{h_3} 9 \xleftarrow{h_7} 7 \xleftarrow{h_9} 3 \xleftarrow{h_7} 9 \xleftarrow{h_3} 3 \xleftarrow{h_9} 7 \xleftarrow{h_3} 9 \xleftarrow{h_7} 7 \xleftarrow{h_9} 3 \dots$$

Translating back into Rosen’s original terminology, we have here  $a = 3$ ,  $b = 7$ ,  $f = 9$ ,  $\Phi = 7$ , but  $\beta = (ev_b)^{-1} = 3$ , the inverse of  $b$ . So  $\beta$  may be reasonably identified with  $b^{-1}$  but not with  $b$ , as pointed out in Cárdenas et al. (2010).

**A linear avatar of Ouroboros.** We sketch here a linear example where sets of “metabolites” are vector spaces instead of integers modulo  $m$ , so that structure preserving mappings are linear. We denote by  $M_{m,n}$  the set of all real matrices with  $m$  rows and  $n$  columns, identified as usual with linear mappings from  $\mathbb{R}^n$  to  $\mathbb{R}^m$ . We put

$C_0 = \mathbb{R}^2 = M_{2,1}$ ,  $c_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ ,  $C_1 = \mathbb{R} = M_{1,1}$  and  $\Phi_0 = \begin{pmatrix} 1 & 0 \end{pmatrix}$  (the first projection of  $\mathbb{R}^2$  onto  $\mathbb{R}^1$ ). Then we find recursively

$c_1 = 1$ ;  $C_2 = H(C_0, C_1) = M_{1,2} \simeq \mathbb{R}^2$ ;  $c_2 = \Phi_0 = \begin{pmatrix} 1 & 0 \end{pmatrix}$ ;  $\Phi_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ , since  $\Phi_1(c_1) = \Phi_0$ ;

$C_3 = H(C_1, C_2) = M_{2,1} \simeq \mathbb{R}^2$ ;  $c_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ ;  $\Phi_2 = Id_2 \in M_{2,2}$  or any matrix with first column  $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ ;

$C_4 = H(C_2, C_3) = M_{2,2} \simeq \mathbb{R}^4$ ;  $c_4 = \Phi_2$ ,  $\Phi_3$  being any matrix with first column  $\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$  if  $\Phi_2 = Id_2$ ;

$C_5 = H(C_3, C_4) = M_{4,2} \simeq \mathbb{R}^8$ , and so on, where we identify matrices with row or column vectors reading their coefficients as usual text. Notice the recursive multiplicative Fibonacci rule  $d_{n+1} = d_n \cdot d_{n-1}$  for  $d_n = \dim C_n$ .

Notice that Rosen’s demanding assumption on the invertibility of the evaluation at  $b (= c_1)$  is satisfied in the arithmetical realization above, where in fact *all* evaluation maps are invertible. In the linear example, the map  $ev_{c_1}$  is still invertible, although the subsequent evaluation maps are not. In particular, any  $2 \times 2$  matrix with first column  $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$  would do as  $\Phi_2$ .

## Ouroboros in Autocatalytic Sets

Here we will approach Ouroboros equation in the spirit of Jaramillo et al. (2010), where attempting to relate the theories of  $(M,R)$  systems and Replicative Autocatalytic Sets (Hordijk and Steel, 2004), a framework for treating molecules as operators was proposed. We will use here the term “metabolism” as synonym of “metabolic network”.

We look upon a metabolism as a directed graph  $\mathcal{M}$  whose set of nodes  $P(X)$  is the collection of all subsets of the set  $X$  of all metabolites and catalysts involved in the metabolism and whose set of arrows  $R$  is given by the reactions  $A \rightarrow B$  in the metabolism ( $A, B \subset X$ ). Molecules  $x$  in  $X$  not produced by the metabolism are coded as reactions of the form  $\emptyset \rightarrow x$ , where the empty set symbol  $\emptyset$  stands for the environment seen as a virtual molecule. We assume further that every metabolite  $x \in X$  appears in the target of some arrow in  $\mathcal{M}$ . Catalysts are defined by a map  $C : R \rightarrow X$  that assigns a molecular identity to the catalyst of each reaction in  $R$ . Of course, we assign the empty catalyst  $\emptyset$  to any arrow (reaction) with source  $\emptyset$ .

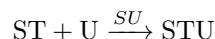
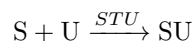
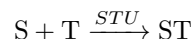
A *premetabolism*  $\mathcal{M}'$  of the metabolism  $\mathcal{M}$  is generated by a subset  $X' \subset X$ , by taking  $P(X')$  as the set of nodes of  $\mathcal{M}'$  and all arrows in  $\mathcal{M}$  whose source lies in  $P(X')$ , as its set of arrows.

There is now a natural sense in which a premetabolism  $\mathcal{M}'$  may be applied to itself, giving raise to a new premetabolism noted  $\mathcal{M}' \circ \mathcal{M}'$ : look at  $\mathcal{M}'$  and just carry out every possible reaction indicated by  $\mathcal{M}'$ ; then collect all the resulting metabolites together to form the metabolite set  $X''$  of the premetabolism  $\mathcal{M}' \circ \mathcal{M}'$ .

Ouroboros avatar in this context reads then

$$\mathcal{M}' \circ \mathcal{M}' = \mathcal{M}'$$

To illustrate this formalism let us introduce a simple molecular system which is an  $(M,R)$  system and a Replicative Autocatalytic Set, taken from Letelier et al. (2006):



This defines a metabolism  $\mathcal{M}$  based on  $X = \{S, T, U, ST, SU, STU\}$ , with  $R$  and  $C$  given by the three reactions above together with  $\emptyset \xrightarrow{\emptyset} S$ ,  $\emptyset \xrightarrow{\emptyset} T$ ,  $\emptyset \xrightarrow{\emptyset} U$ . Now, writing just  $X'$  for a premetabolism  $\mathcal{M}'$ , we can calculate for instance:

$$\{S, T, SU, STU\} \circ \{S, T, SU, STU\} = \{S, T, ST\},$$

$$\{S, T, ST\} \circ \{S, T, ST\} = \{S, T\},$$

$$\{S, T\} \circ \{S, T\} = \{S, T\},$$

so this premetabolism dies out to a trivial solution of Ouroboros equation (i.e. one whose associated reactions are all of the form  $\emptyset \xrightarrow{\emptyset} x$ ). On the contrary, we have  $\{S, T, U, SU, ST, STU\} \circ \{S, T, U, SU, ST, STU\} = \{S, T, U, SU, ST, STU\}$ ,

i.e.  $\{S, T, SU, ST, STU\}$  defines a non trivial solution to Ouroboros equation!

## Ouroboros in Autopoietic systems

Before concluding we would like to bring in the theory of Autopoiesis, as it has deep connections to the idea of

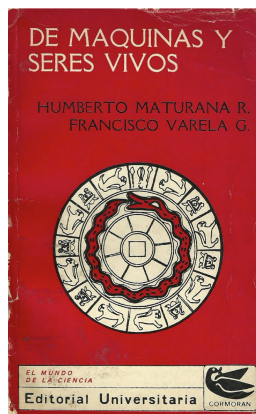


Figure 1: Original cover of the book introducing autopoietic systems (Maturana and Varela, 1973). Although the notion of self-reference is not explicitly mentioned in the book, the authors chose an Ouroboros to illustrate its cover.

self-reference. In fact one of its creators, Varela spent almost a decade looking for a suitable framework to formalize the notions behind this connection (Varela, 1975; Varela and Goguen, 1978; Varela, 1981; Soto-Andrade and Varela, 1984). We won't attempt to reproduce his results, but instead show why self-reference arises from the conceptualization of Autopoiesis theory.

First, we should introduce the perspective of Maturana and Varela for defining a system. A system (or machine) is defined as a unity distinguishable from its surroundings, characterized by two concepts: *organization* and *structure*. The former relates to all processes (or relations) that define the system as a unit and that determine the dynamics of transformations and interactions that the system may undergo as such a unit. The latter are all actual relations that hold between the components of the system in a given space and time (Maturana and Varela (1980), pages 77-84). Now we can define an Autopoietic system (*loc. cit.*) as a network of processes of production, such that its components satisfy the following:

- i) through their interactions and transformations regenerate and realize the network of processes that produced them;
- ii) constitute the system as a concrete unity in the space in which the components exist by specifying the topological domain of its realization.

The first property of Autopoietic systems can be interpreted as a description of a closed network of production or *metabolic closure*, where the elements needed for the occurrence of each step of the network (such as catalysts) are produced by the network itself. From a dynamical perspective, can also be viewed as non trivial fixed-points of the network dynamics. The notion of *metabolic closure* is com-

mon and comparable between several theories of living systems (see Cárdenas et al. (2010) for references). However, Autopoiesis demands more than self-production. What is maintained and reconstituted through the system's dynamics is its *organization*, i.e. what makes it distinguishable as a unit. This is secured in time by the first property and in space by the second. Therefore, if we were to define an autopoietic system we would be tempted to say something like "a unit that regenerates what distinguishes itself as a unit...".

As the last idea suggests, *organizational invariance* can be understood as an ultimate case of recursion or self-reference. In the previous sections we have discussed how to find consistent and non trivial cases where self-reference is possible; future challenges would involve bringing both properties of autopoietic systems into our framework.

## Conclusion and final remarks

As we have surveyed here,  $f(f) = f$  is an intriguing equation that abstracts phenomena from many fields. It must be underlined that our interest in this topic arose from a very basic (and unsolved) question in theoretical biology: "What is a correct theoretical framework to formalize systems that construct themselves?". Metabolism is an outstanding example, as the action of metabolism results in the reconstitution of the components that were responsible for its occurrence in the first place.

We are of the opinion that, in order to construct a formalism that captures Metabolism from the perspective of Autopoiesis and  $(M,R)$ -systems, self-reference is an unavoidable point to consider - not to be confused with simulations of Metabolism, which we regard as complementary efforts. As presented in this paper, dealing with self-reference mathematically, even if it seems to challenge our classical conceptions, is certainly feasible.

Nevertheless, we are aware that the methods exposed are still halfway towards a definitive theory. In particular we should be able to move beyond hypothetical examples into a framework closer to concrete biological systems. Towards this goal there are several avenues for improvement. For instance, so far we have interpreted Metabolism as a network of reactions and catalysis, leaving for later other dimensions of Metabolism, such as time. Self-reference could be regarded as identity conservation under the Metabolic dynamics (Varela, 1975; Varela and Goguen, 1978), and we expect that adding the temporal dimension should allow us to ask more complex questions, closer to molecular systems. Also, we haven't looked closely at the physicochemical properties of Metabolism, which may provide a grounding as well as a guide for our mathematical models.

In another avenue, one of the main lessons is the vanishing dichotomy between operand and operator, implicit in  $f(f)$ . This suggest that the phenomena of interaction more than application (in the old functional sense), or concurrency more than sequentiality, may constitute a more appropriate

metaphor. As it is well known, life phenomena are intrinsically concurrent, and as such, it appears natural that the emerging formalisms for concurrency are beginning to be applied to this field (Milner, 2009; Cardelli, 2005). We wonder whether there may be avatars of Ouroboros lurking in the concurrent world, an interesting question to explore in future work.

## Acknowledgments

We would like to dedicate this paper to the memory and work of Francisco Varela, friend and mentor.

## References

- Aczel, P. (1988). *Non-well-founded sets*, volume 14 of *CSLI Lecture Notes*. Stanford University, Center for the Study of Language and Information, Stanford, CA.
- Barendret, H. (1984). *The Lambda Calculus: Its Syntax and Semantics*. North Holland, Amsterdam.
- Cardelli, L. (2005). Abstract machines of systems biology. In Priami, C., Merelli, E., Gonzalez, P., and Omicini, A., editors, *Transactions on Computational Systems Biology III*, volume 3737 of *Lecture Notes in Computer Science*, pages 145–168. Springer, Berlin.
- Cárdenas, M. L., Letelier, J. C., Gutierrez, C., Cornish-Bowden, A., and Soto-Andrade, J. (2010). Closure to efficient causation, computability and artificial life. *J. Theor. Biol.*, 263:79–92.
- Church, A. (1951). *The Calculi of Lambda Conversion*, volume 6 of *Annals of Math. Studies*. Princeton.
- Hordijk, W. and Steel, M. (2004). Detecting autocatalytic, self-sustaining sets in chemical reaction systems. *J. Theor. Biol.*, 227:451–461.
- Jaramillo, S., Honorato-Zimmer, R., Pereira, U., Contreras, D., Reynaert, B., Hernández, V., Soto-Andrade, J., Cárdenas, M., Cornish-Bowden, A., and Letelier, J. (2010).  $(M,R)$  systems and raf sets: common ideas, tools and projections. In Feller-mann, H., Dörr, M., Hanczyc, M., Laursen, L., Maurer, S., Merkle, D., Monnard, P., Stoy, K., and Rasmussen, S., editors, *Artificial Life XII: Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*, pages 94–100.
- Kampis, G. (1995). Computability, self-reference, and self-amendment. *Communication and Cognition - Artificial Intelligence*, 12:91–109.
- Kauffman, L. H. (1987). Self-reference and recursive forms. *J. Social and Biological Structures*, 10:53–72.
- Letelier, J., Kuboyama, T., Yasuda, H., Cárdenas, M., and Cornish-Bowden, A. (2005). A self-referential equation,  $f(f) = f$ , obtained using the theory of  $(M,R)$  systems: overview and applications. In Anai, H. and Horimoto, K., editors, *Algebraic Biology 2005*, pages 115–126. Universal Academy Press, Tokyo.
- Letelier, J. C., Soto-Andrade, J., Guíñez Abarzúa, F., Cornish-Bowden, A., and Cárdenas, M. L. (2006). Organizational invariance and metabolic closure: analysis in terms of  $(M,R)$  systems. *J. Theor. Biol.*, 238:949–961.
- Löfgren, L. (1968). An axiomatic explanation of complete self-reproduction. *Bulletin of Mathematical Biology*, 30:415–425.
- Maturana, H. and Varela, F. (1973). *De Máquinas y Seres Vivos*. Editorial Universitaria, Santiago, Chile.
- Maturana, H. and Varela, F. (1980). *Autopoiesis and Cognition: the Realisation of the Living*. D. Reidel Publishing Company, Dordrecht, The Netherlands.
- Milner, R. (2006). *Turing, Computing, and Communication*, pages 1–8. Springer, Berlin.
- Milner, R. (2009). *The Space and Motion of Communicating Agents*. Cambridge University Press.
- Quine, W. V. (1980). *New Foundations for Mathematical Logic*, pages 80–101. Harvard Univ. Press, 2nd edition.
- Rosen, R. (1958). The representation of biological systems from the standpoint of the theory of categories. *Bulletin of Mathematical Biology*, 20(4):317–341.
- Rosen, R. (1959). On a logical paradox implicit in the notion of a self-reproducing automaton. *Bull. Math. Biophys.*, 21:387–394.
- Rosen, R. (1991). *Life Itself*. Columbia University Press, New York.
- Scott, D. S. (1972). Continuous lattices. In Lawvere, F. W., editor, *Toposes, algebraic geometry and logic (1971 Dalhousie University Conference)*, volume 74 of *Lecture Notes in Mathematics*, pages 97–136. Springer-Verlag, New York.
- Scott, D. S. (1973). Models for various type-free calculi. In Suppes, P., Henkin, L., Joja, A., and Moisil, G. C., editors, *Logic, Methodology and Philosophy of Science IV: Proceedings of the Fourth International Congress*, volume 74 of *Studies in Logic and the Foundations of Mathematics*, pages 157–188. North-Holland, Amsterdam.
- Soto-Andrade, J. and Varela, F. J. (1984). Self reference and fixed points. *Acta Appl. Mathematica*, 2:1–19.
- Spencer Brown, G. (1969). *The laws of form*. George Allen and Unwin, London.
- Varela, F. (1975). A calculus for self-reference. *International Journal of General Systems*, 2(1):5–24.
- Varela, F. (1979). The extended calculus of indications interpreted as a three-valued logic. *Notre Dame J. Formal Logic*, 20:141–146.
- Varela, F. (1981). Autonomy and autopoiesis. In Gerhard, R. and Schwegler, H., editors, *Self-organizing systems: An interdisciplinary approach*, pages 14–23. Springer-Verlag, New York.
- Varela, F. and Goguen, J. A. (1978). The arithmetic of closure. *Journal of Cybernetics*, 8:291–324.
- Wittgenstein, L. (1961). *Tractatus Logico-Philosophicus*. Routledge and Kegan, London, english edition.

# Recruitment, Selection and Alignment of Spatial Language Strategies

Michael Spranger<sup>1</sup>

<sup>1</sup>Sony CSL Paris, 6 rue Amyot, 75005 Paris, France  
spranger@csl.sony.fr

## Abstract

All languages of the world have a way to talk about space and spatial relations of objects. Cross-culturally, immense variation in how people conceptualize space for language has been attested. Different spatial conceptualization strategies such as *proximal*, *projective* and *absolute* have been identified to underlie peoples conception of spatial reality. This paper argues that spatial conceptualization strategies are negotiated in a cultural process of linguistic selection. Conceptualization strategies originate in the cognitive capabilities of agents. The ecological conditions and the structure of the environment influence the conceptualization strategy agents invent and which corresponding system of lexicon and ontology of spatial relations is selected for. The validity of these claims is explored using populations of humanoid robots.

## Introduction

Human language is a complex adaptive system (Beckner et al., 2009), which is shaped by its users in a process of cultural evolution in order to achieve communicative goals such as drawing the attention to an object in the environment using spatial language. Language evolves constrained by factors such as communicative success, expressivity, learnability and ecological significance. This paper argues that these claims are also true for spatial language and that they are at the heart of explanations for the diversity of spatial language attested across different cultures.

Spatial language exhibits enormous amount of cross-cultural variation on two levels.

**Spatial language systems** Spatial language is typically a conglomerate of different systems. English for instance has a *proximal* system consisting of the two spatial relations “near” and “far”, a *projective* system including relations such as “left” and “front”. Moreover, English features an *absolute* system of spatial relations, e.g. “north” and “east”. Languages differ with respect to the particular organization of language systems. Spanish, for instance, features three proximal relations (Kemmerer, 1999).

**Spatial language strategies** Languages differ qualitatively in the kind of systems they support. For instance, some

languages such as the Mayan language Tenejapan do not have projective terms but only absolute spatial relations (Levinson, 2003). Speakers of this language conventionally refer to objects in the immediate vicinity as *uphill* or *downhill*. Tenejapan speakers, therefore, habitually conceptualize reality differently than speakers of English.

There are two questions immediately following from this observation: (1) how do language systems form, (2) what are the origins of strategies. If one wants to study the evolution of spatial language, answers to the origins and development of both layers of language change have to be identified. Previous work has shown how language strategies can form language systems, e.g. for color and actions (see Steels, 2011 for an overview). In these experiments, agents are a priori endowed with a particular language strategy which includes a way of construing reality plus a battery of language change operators. The experiments then show that given these prerequisites autonomous agents can negotiate a particular system of categories (ontology) and words (lexicon).

Recently the origins of language strategies themselves have come under investigation. Bleys (2010) proposes that color strategies are under selective pressure driven by communicative success and cognitive effort (see also van Trijp, 2010 for a similar argument). This paper broadens this approach by extending it to spatial language and, most importantly, by proposing a concrete account of the origins of language strategies. Three important concepts guide our discussion (Steels, 2011).

**Recruitment** Language strategies are grounded in general cognitive capabilities and operations (Steels, 2007). For instance, the absolute strategy in English requires that agents are able to categorize objects using spatial categories that relate to particular geocentric features of the environment. In English absolute system this is related to compass readings and map use (Tenbrink, 2007). In other languages such features can include geocentric landmarks such as mountains which are always visible, or other global features such as the aforementioned uphill-downhill distinctions (Levinson, 2003). The categorization of these objects themselves is a cognitive ability that

needs to be present before a linguistic absolute spatial system can form. Cognitive operations are recruited and assembled to form spatial conceptualization strategies.

**Selection** Once a strategy has formed it is used to build a concrete system of spatial categories and linguistic means to express them. For instance, in the simplest case a strategy is expressed lexically by naming the spatial relations. The system and the strategy are both subject to selective pressures. Other strategies might compete in terms of success, expressivity and ecological significance. To organize competition and selection, the overall success of a strategy and the associated ontology and lexicons are tracked.

**Alignment** Language is a phenomenon that occurs in the interactions of individuals of a group of language users. Language strategies or any linguistic material are invented in local interactions in which typically few members of a population participate. Different parts of the population might invent other strategies. This poses a problem as for language to be usable it needs to be conventionally used and known to the complete population. Alignment is the process by which a strategy and the corresponding language systems spread in the population. We organize alignment of strategies using the scoring of strategies used for orchestrating selection and competition.

This paper gives a mechanistic account of the origins and evolution of spatial language strategies by identifying concrete cognitive operations, selection and alignment mechanisms. We defend the main claim using artificial language evolution experiments which have been a key technique to identify, explore and validate ideas about cultural language evolution (Steels, 1995; Kirby, 2002; Smith et al., 2003).

## Adaptive Spatial Language Games

For researching the basic claims of this work, we setup experiments in which robotic agents (Sony humanoid robots, see Fujita et al., 2003) encounter objects in spatial scenes. Such setups are called *spatial language games* and they package a specific intention – talking about objects in the environment – with a specific interaction script.

Figure 1 shows the environment in which two robots interact. Both robots are equipped with a vision system that singles out and tracks objects (Spranger, 2008). The environment contains four types of objects: *blocks*, *boxes*, *robots* and *geocentric markers*. The vision system extracts the objects from the environment and computes a number of raw, continuous-valued features such as *x*, *y*, *width*, and *height*, but also color values in the YCrCb color space.

Always two agents randomly drawn from a population interact, one acts as the speaker, the other as the hearer. The spatial language game uses the following game script assum-

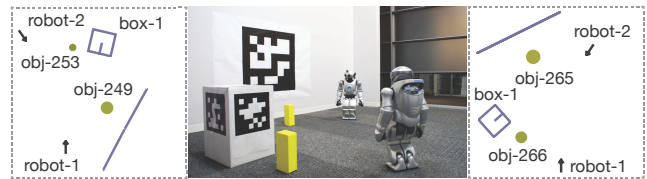


Figure 1: Spatial setup. To the left the world model extracted by the left robot is shown. To the right the same for the other robot is depicted.

ing a population  $P$  of agents, and a world consisting of a set of individual objects.

1. The speaker selects an object out of the context, further called the topic  $T$ .
2. The speaker tries to find a meaning comprised of a particular spatial relation and a particular way of conceptualizing reality for describing the topic.
3. The speaker looks up the word associated with the spatial relation in his memory and produces the word.
4. The hearer looks up which relation is associated with this word in his memory and examines the context to find a unique object which satisfies the relation.
5. The hearer points to this object.
6. The speaker checks whether the hearer selected the same object as the one he had originally chosen. If they are the same, the game is a *success* and the speaker signals this outcome to the hearer.
7. If the game is a *failure*, the speaker points to the topic  $T$  he had originally chosen.

Such an interaction can fail for different reasons. For instance, the speaker might be unable to discriminate the topic object because he is missing a spatial relation or a conceptualization strategy. Both success and failure of communication provide opportunities agents to adapt their linguistic knowledge, ontologies and repertoires of conceptualization strategies.

## Grounded Spatial Conceptualization Strategies

We use a computational formalism called *Incremental Recruitment Language* (IRL) that was specifically developed for representing adaptive conceptualization strategies (Spranger et al., 2010) and spatial semantics. To make this more concrete let us consider the semantics underlying a specific spatial phrase. Figure 2 shows the representation of the spatial semantics of a phrase like “near the box” which consists of a spatial relation (near) plus additional information about the landmark (the box).



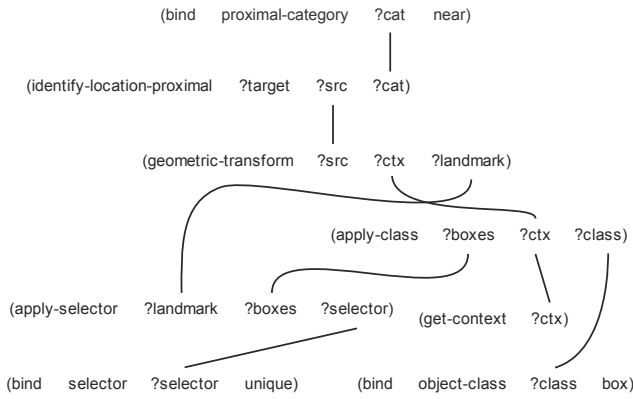


Figure 2: IRL-program representing the semantic structure of the phrase “near the box”.

The main idea behind IRL is that semantic structure is procedural (Johnson-Laird, 1977) and can be represented using programs (IRL-programs). Consequently, we represent the semantics of the phrase as a set of cognitive operations such as applying a categorization (`identify-location-proximal`), and transform the viewpoint on the scene to a specific object (`geometric-transform`) that are linked in a certain way. For instance, the output of the operation `geometric-transform` linked by the variable `?src` (all variables start with a `?`) is connected to the input of the categorization operation. In other words, once the set of objects from the context (introduced by `get-context`) is transformed to a particular viewpoint then the spatial category is applied.

The following operations (excerpt) are used as building blocks for spatial conceptualization strategies.

**geometric-transform** transforms the environment to a particular landmark object (in the example this is the box).

**identify-location-proximal** applies the spatial category given as argument to the input source set. The operation returns the single object which has the highest similarity with the spatial category. This operation applies proximal relations.

**identify-location-projective** works similar to the previous operation but is special to projective relations. We use the intrinsic notion of projective relations (Levinson, 2003). Landmarks such as the box and the robots can have an inherent orientation which highlights one of their sides as being the front.

**identify-object-absolute** encodes an absolute strategy. Absolute strategies compute rotation based on the direction towards a geocentric wall marker available in some spatial scenes.

Besides cognitive operations (algorithms), semantic structure also contains data. So called `bind`-statements introduce pointers to agent internal representations of concepts, prototypes and spatial relations. For example, `(bind proximal-category ?cat near)` introduces the spatial category `near`. Spatial relations are implemented using insights from cognitive semantics (Herskovits, 1986) and prototype theory (Rosch, 1975). There are two types of categories, distance-based (proximal) and angle-based (projective and absolute).

**Angular relations** Angular categories (projective and absolute relations) have a focal region around a specified axis. Similarity of some location to an angular category depends on the distance of angles. For instance, the front category has a high degree of applicability along the frontal axis. The following equations defines the degree of applicability, i.e. similarity,  $\text{sim}_a \in [0, 1]$  given an object  $o$  and an angular category  $c$  and a parameter  $\sigma$  which steers the steepness of the function.

$$\begin{aligned}\text{sim}_a(o, c) &:= e^{-\frac{1}{2\sigma c} d_a(o, c)} \\ d_a(o, c) &:= |a_o - a_c|\end{aligned}$$

$a_o$  denotes the angle of the position of  $o$  to the coordinate center and  $a_c$  is the prototypical angle of  $c$ .

**Proximal relations** Proximal relations are represented using prototypical distances.

$$\begin{aligned}\text{sim}_d(o, c) &:= e^{-\frac{1}{2\sigma c} d_d(o, c)} \\ d_d(o, c) &:= |d_o - d_c|\end{aligned}$$

$d_o$  denotes the distance of the object  $o$  to the coordinate center and  $d_c$  is the prototypical distance of the proximal category  $c$ .

**Spatial conceptualization strategies** The IRL-program Figure 2 shows a specific semantic structure that is part of a specific conceptualization strategy, namely the proximal spatial strategy. If we remove the spatial relation from the IRL-program in that figure, we are left with a conceptualization strategy which involves a landmark (the box) and a (unspecified) proximal spatial relations. We call such partial structures *chunks* (Spranger et al., 2010). Chunks are reified conceptualization strategies. They have a score which represents how much the agent prefers the strategy over others (e.g., see Mainwaring et al., 2003 for preferences in perspective choice).

Spatial conceptualization strategies involve more than just a choice of spatial relations. Landmarks, perspective, frames of reference (Tenbrink, 2007) are all important aspects of the construal of spatial relations and researchers are still mapping out the taxonomies and unifying theories for the vast



amount of spatial conceptualization strategies found in natural language (Levinson, 2003). For instance, which landmarks can be used with a particular spatial relation – just people, animals or also inanimate objects – is part of the choices manifest in a particular strategy. We can represent all these different factors using distinct cognitive operations and IRL-programs.

**Production and interpretation** When agents communicate they face the problem which language strategy to choose: proximal, projective or absolute. Within each strategy there are additional choices which spatial relation the agent wants to use, and which landmark to employ. Finally, agents have to name the category and retrieve a name for it in order to make themselves understood<sup>1</sup>. Production – the process of finding an utterance for discriminating an object – and interpretation – the process of finding the topic given an utterance – are heuristics guided, automated search processes that try to find good semantic structure (IRL-programs).

**Production** In production, agents choose the spatial conceptualization strategy and the spatial relation which is most *discriminating* the topic T with respect to all other objects in the context. A strategy and the chosen category are discriminating if they maximize the similarity of the topic but minimize the similarity of all other objects (Herskovits, 1986). Once the category is chosen, agents will verbalize the category by retrieving the term associated with the category.

**Interpretation** In parsing, this process is reversed and agents use their lexicon to find the category linked to the spatial term in the utterance. The category is used to find back the conceptualization strategy which is in turn applied together with the spatial relation to single out the topic.

We use *Fluid Construction Grammar* (FCG) (Steels and De Beule, 2006) for verbalization. FCG is a formalism developed for language evolution in which linguistic knowledge is represented using form-meaning associations, so called *constructions*. Constructions are scored and can be freely and deleted from an agent's memory which allows to model the change of linguistic knowledge of that agent.

Constructions are not the only items that are scored. Production and interpretation are heavily influenced by the score of the different linguistic items. Spatial relations, conceptualization strategies (chunks) and lexical items all have individual scores associated with them which are used to weight the results. The scores reflect individual preferences.

<sup>1</sup>In this paper, agents are confined to uttering single words in spatial language games.

## Co-Evolution of Spatial Relations

Conceptualization strategies are necessary prerequisites for building ontologies and lexicons. This section shows that given a chunk and a set of invention, adoption and alignment operators concrete systems of spatial relations can be negotiated in populations. Due to space constraints this section only exercises this for the projective strategy. Similar propositions hold for absolute and proximal strategies (Spranger, 2011b). The following paragraphs detail the operators.

**Invention:** *Speaker cannot find a discriminating spatial category in production*

- **Diagnostic:** When the speaker cannot conceptualize a meaning (step 2 of the spatial language game fails).
- **Repair:** The speaker constructs a spatial relation  $R$  based on the relevant strategy (projective) and the topic pointed at. The new category is necessary based on the distance or angle observed for the topic object (the initial sigma is small 0.1). Additionally, the speaker invents a new construction associating  $R$  with  $s$ .

**Adoption:** *Hearer encounters unknown spatial term  $s$*

- **Diagnostic:** When the hearer does not know a term (step 3 fails).
- **Repair:** The hearer signals failure and the speaker points to the topic T. The hearer then constructs a spatial relation  $R$  based on the relevant strategy and the topic pointed at. Additionally, the speaker invents a new construction associating  $R$  with  $s$ .

**Category alignment** Projective categories are represented by prototypical angles. After each interaction agents update the prototypical angle to better reflect the new observation by averaging the angles of objects in the sample set  $S$ . The new prototypical angle  $a_c$  of the category is computed using the following formula for averaging angles.

$$a_c = \text{atan2} \left( \frac{1}{|S|} \sum_{s \in S} \sin a_s, \frac{1}{|S|} \sum_{s \in S} \cos a_s \right) \quad (1)$$

The new  $\sigma$  value  $\sigma'$  which describes the shape of the similarity function of the category is adapted using the following formula.

$$\sigma'_c = \sigma_c + \alpha_\sigma \cdot \left( \sigma_c - \sqrt{\frac{1}{|S| - 1} \sum_{s \in S} (a_c - a_s)^2} \right) \quad (2)$$

This formula describes how much the new  $\sigma_c$  of the category  $c$  is pushed in the direction of the angle standard deviation of the sample set by a factor<sup>2</sup> of  $\alpha_\sigma \in [0, \infty]$ .

<sup>2</sup> $\alpha$  is given by the experimenter and in all experiments described here  $\alpha = 0.5$

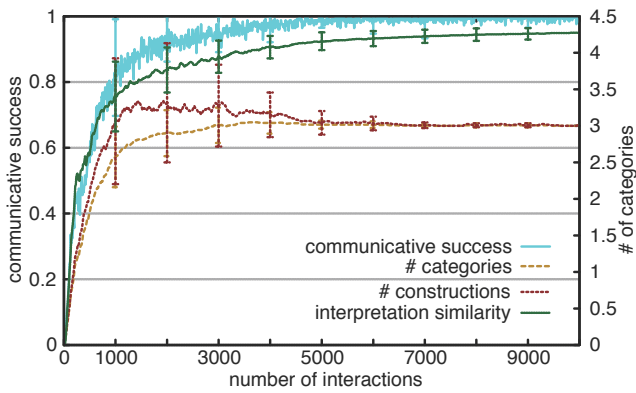


Figure 3: Results for a formation experiment in which agents develop a projective category system.

**Lexicon alignment** The invention and adoption repairs introduce a particular problem – the problem of *synonymy*. Synonymy occurs when an agent explicitly represents that a spatial category can be named using different spatial terms. Each of these different names is represented using a separate construction each of which links the synonymously used category to a different string. Allowing agents to track synonymy in their lexicons can be beneficial for overall lexicon size, but only if agents also have additional mechanisms for managing synonymy. Such a mechanism, called *lateral inhibition*, was introduced in Steels (1995):

- In case the interaction was a success both speaker and hearer reward the winning construction – the one used in production and interpretation – by a score of  $\delta_{\text{success}}$ . Competing constructions are punished by  $\delta_{\text{inhibit}}$ . There are two types of competing constructions. First, there are those constructions which associate the same spatial relation but with a different word. Second, there are constructions that link the same word to different spatial relations.
- After a failed game, both speaker and hearer decrease the score of the used association with  $\delta_{\text{fail}}$ .

**Measures** To be sure that our approach to formation works reliably, we test it by running multiple trials of the same experiment. In each trial agents start with an empty ontology and lexicon. Success, performance and language development of the population are tracked using the following measures.

**Communicative Success** Communicative success is the most important measure as it reflects the overall performance of the population. Every interaction is either a success or a failure. Success is counted with 1.0 and failure is counted as 0.0.

### Number of Categories and number of constructions

This measure simply counts the average number of categories and constructions known to the agent.

**Interpretation Similarity** This is a measure tracking how similar the interpretation of each word known to each agent is. For this the categories attached to the word in each agent is compared. Since projective categories are described by a direction and a similarity function width parameter  $\sigma$ , two categories are most similar (1.0) when both angle and  $\sigma$  are equal.

**Results** Figure 3 shows the dynamics of experiments in which 10 agents start without any categories and constructions and gradually have to solve their communicative problems by invention and adoption of linguistic and semantic material (25 trials). In each trial 10000 spatial language games are played, with two agents randomly drawn from the population, interacting, and inventing, adopting and aligning linguistic knowledge.

The graph shows that agents are able to form successful language systems that gradually become more and more similar in the population as the linguistic knowledge spreads from agent to agent. After 10000 interactions agents are communicating successfully in over 95% of the interactions. In all trials, the population agrees on using a total of three spatial relations and corresponding names.

## Selection and Alignment of Spatial Conceptualization Strategies

The previous section demonstrated that given a conceptualization strategy and strategies for invention, adoption and alignment agents can co-evolve successful systems for referring to objects in their environment. The important claim in this section is that conceptualization strategies are negotiated in a cultural process, similar to how the lexicon is negotiated, through local interactions by agents in a community. The idea is that a particular strategy survives when it is relevant to an agent because it is efficient and useful in discriminating objects and it contributes to the communicative success of an agent at least in a few spatial contexts.

**Selection and Alignment** Selection of a strategy is intricately linked to the success of the ontology and lexicon, i.e. spatial category system, it builds. For instance, if an agent is building a language system with an absolute strategy this entails that the absolute relations and the strategy itself are subject to the same selective pressure. It is the success of the overall system, i.e. the spatial relations together with the performance of the strategy, that drives the organization of the syntactic and semantic repository of the agent.

The previous section talked about the invention and alignment of words and categories. The same operators are used for building different language systems. Additionally, the

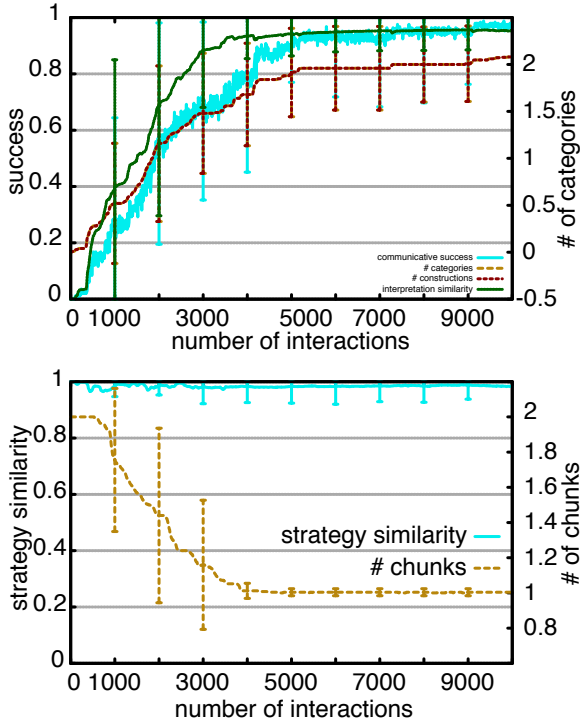


Figure 4: Dynamics of a category formation experiment in which 10 agents align the conceptualization strategy used at the same time.

success of a strategy, i.e. chunk, is tracked after every interaction by updating its score. If the conceptualization strategy was used successfully its score is increased by a factor  $\delta_{\text{success}}$  otherwise it is punished by  $\delta_{\text{failure}}$ . All other conceptualization strategies not used are punished by the score  $\delta_{\text{competitor}}$ . The value of these deltas is typically by a magnitude lower than the deltas for updating categories and words.

**Measures** We test our approach by running experiments in which agents are given different conceptualization strategies. To monitor the alignment of conceptualization strategies we use an additional measure.

**Number of chunks** This measure averages the number of conceptualization strategies with a score bigger than 0 over every agent.

**Conceptualization strategy similarity** The *css* is defined for a population  $P$  as the average *acss* for every two agents. Since *acss* is symmetric, all combinations of two agents are considered.

**Agent conceptualization strategy similarity** The *acss* is computed by comparing the score of each strategy. Since strategies are never removed but merely reduced to a score of 0.0 we can compute a distance of scores between the

chunks in each agent and envelope the result using an exponential decay function which results in the following formula.

$$\text{acss}(a_1, a_2, S) := e^{(-1 \cdot \sum_{s \in S} |\text{score}(s, a_1) - \text{score}(s, a_2)|)}$$

In this formula  $a_1, a_2$  are the agents whose similarity score is computed,  $S$  is the set of strategies given to agents and  $\text{score}(s, a_1)$  is the score agent  $a_1$  gives to strategy  $s$ .

**Experimental Setup and Results** We test the power of strategy alignment using contexts which can be manipulated to feature absolute and intrinsic properties. More specifically, we manipulate the distribution of intrinsic and absolute properties in the environment. Figure 4 shows the dynamics of an experiment where agents start equipped with two strategies: an absolute and an intrinsic one. The environment is such that it favors absolute systems. In 50% of the scenes both intrinsic and absolute features are present. In the remaining 50% of the contexts only absolute features are present and no intrinsic ones.

The environmental conditions have a strong effect on the development of the system. All 25 populations agree on using an absolute strategy. What is important is that the contexts where only absolute features are present reward the absolute strategy and punish the intrinsic conceptualization strategy. Consequently, even in a context where intrinsic and absolute features are present, the absolute strategy is preferred. The development of such a preference has important effects on the invention of categories. Because of the preference for the absolute strategy, invention of categories shifts to producing only absolute categories. The successful use of these categories enforces the absolute strategy and leads to further punishment of the intrinsic strategy. The effect is that only the absolute strategy survives. Additionally, the graph shows that roughly together with the category system, agents align their conceptualization strategy.

## Recruitment of Conceptualization Strategies

Conceptualization strategies are networks of cognitive operations encoding a particular way of construing reality. Consequently, they originate in a process of *recruitment* which assembles cognitive operations into strategies, i.e. chunks. Recruitment is a necessary pre-requisite for the usage of conceptualization strategies and their alignment in a population. Once a chunk is invented it immediately extends the conceptualization capabilities of the inventing agent.

**Recruitment** Strategy invention is deeply integrated into the processing of agents. Agents unable to conceptualize or unable to conceptualize with sufficient confidence diagnose a problem which is fixed by a repair that starts the search for new conceptualization strategies. The reason for this integration specifically with other invention mechanisms such as

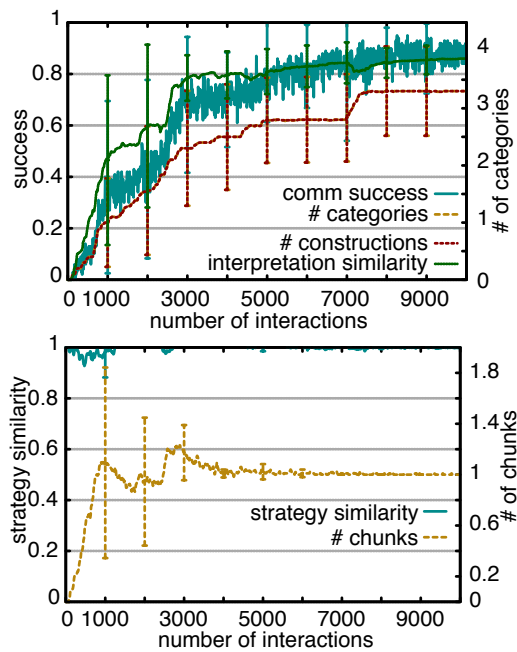


Figure 5: Results for strategy invention, alignment and category development. A population of 10 agents develops both conceptualization strategies as well as lexical systems for spatial strategies corresponding to these strategies.

category invention is that agents when inventing new strategies also immediately have to invent new categories with these strategies because a strategy itself is not verbalized but the name of the spatial relation. This sort of dual invention is especially important in the beginning of experiments, when agents have neither developed strategies nor categories.

But there is a second reason for deep integration of strategy invention. When an agent already has developed a strategy then he might also solve a particular communicative problem by inventing new categories for established strategies. Such decisions whether to use a new category with an existing strategy or a new strategy with an existing category, or even to use a newly invented strategy with a newly invented category are made based on the discriminative power of each these different possibilities in the particular context. So for instance if an existing strategy has a low score the probability of inventing a new strategy increases, whereas if the current topic can be sufficiently discriminated using an existing strategy no invention occurs.

We need two more operators besides the operators discussed in previous sections.

**Invention:** *Speaker cannot find a meaning for referring to the topic*

- **Diagnostic:** When the speaker cannot conceptualize a meaning (step 2 of the spatial language game fails).

- **Repair:** The speaker invents new conceptualization strategies by assembling cognitive operations such as identify-proximal, geometric-transform into chunks which is immediately followed by the invention of categories for each new chunk (see section on co-evolution of categories and terms). At this point the speaker might have a number of new solutions to his conceptualization problem consisting of new strategies and new corresponding spatial relations. Subsequently, the speaker selects the strategy and category which is *most discriminating*. Once selected, he invents a new word and construction for expressing the new strategy.

**Adoption:** *Hearer encounters unknown spatial term  $s$*

- **Diagnostic:** When the hearer does not know a term (step 3 fails).
- **Repair:** The hearer signals failure and the speaker points to the topic  $T$ . The hearer then constructs new strategies, i.e. chunks, and for each of them he invents a new spatial relation  $R_i$  based on the the topic pointed at. The hearer then decides on which of the strategies is most discriminating. This is the one selected for storing. Additionally, the hearer invents a new construction linking  $R_i$  with  $s$ .

These two invention and alignment operators are specific to the invention of chunks. Moreover, agents are equipped with the selection and alignment operators for chunks, spatial relations and words discussed earlier.

**Results** Figure 5 shows the dynamics of invention and alignment of conceptualization strategies in a population of 10 agents (25 trials). Agents have a repository of 10 basic cognitive operations from which they can draw new building blocks whenever there are problems in communication. They can choose different landmarks: the robot, or the box, and different category systems absolute and intrinsic projective, as well as proximal. The agents manage to agree on one particular strategy while at the same time developing a category system and a lexicon from scratch.

However, the process does not show the same overall success as previously discussed experiments. The reason is that conceptual alignment is a difficult process which is complicated by the number of choices in strategies, population size and the variety of different contexts and discriminative situations which might all favor different strategies. In some contexts proximal is the best strategy, some allow absolute and/or projective categories to be invented. Nevertheless, agents do come to an agreement. Here, they agree on average on a single conceptualization strategy.

For space reasons, we can only discuss one particular experiment with trials all equal in environmental condition. But, of course once the system is setup one can study the effect of varying conditions. The systems discussed here

are very flexible and find solutions to different environmental conditions featuring additional landmarks, intrinsic and absolute features. Additionally, agents react flexibly to different object distributions that favor distance-based or angle-based strategies.

## Discussion

This paper has argued for selection, recruitment and alignment as the basic mechanisms explaining the evolution of language strategies together with corresponding language systems. We have shown (1) how strategies can be represented, (2) how strategies build language systems, (3) how selection works on strategies and (4) how strategies are built by recruiting cognitive operations. We provided mechanistic explanations and validated them in robotic experiments.

The basic claim validated is that we can understand the evolution of strategies as a process of cultural negotiation fueled by the cognitive capabilities of agents, i.e. the cognitive operations available. The process is constrained by environmental factors such as the availability of geocentric landmarks. While cognition and ecology influence the selection process, the negotiation takes place within a single static population via linguistic interactions. This is also the main difference to other models of cultural evolution which claim that intergenerational turnover is the main cause of language change (Kirby, 2002; Smith et al., 2003).

We have only considered a simple lexical verbalization strategy. Certainly, spatial language shows much more variation in the kinds of syntactic material that is employed to convey distinct spatial semantics. A discussion can be found in Levinson and Wilkins (2006) and Tenbrink (2007) and evolutionary models in Spranger (2011a). Moreover, spatial language can feature other conceptualization strategies involving toponyms, directional categories or body-centered spatial relations. Given a suitable implementation of cognitive operations, we claim that the same approach can be used to study the evolution of such strategies.

## Acknowledgements

I am greatly indebted to Masahiro Fujita, Hideki Shimomura, and their team for creating the Sony humanoid robots and for making them available for this research. I thank Luc Steels and Martin Loetzsch for help with the robotic setup, and I thank Kateryna Gerasymova for help with writing the paper. This research was funded by the Sony CSL Paris with additional funding from the EU FP7 ALEAR project.

## References

- Beckner, C., Blythe, R., Bybee, J., Christiansen, M. H., Croft, W., Ellis, N. C., Holland, J., Ke, J., Larsen-Freeman, D., and Schoenemann, T. (2009). Language is a complex adaptive system: Position paper. *Language Learning*, 59:1–26.
- Bleys, J. (2010). *Language Strategies for the Domain of Colour*. PhD thesis, Vrije Universiteit Brussels (VUB), Brussels, Belgium.
- Fujita, M., Kuroki, Y., Ishida, T., and Doi, T. (2003). Autonomous behavior control architecture of entertainment humanoid robot SDR-4X. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 960–967.
- Herskovits, A. (1986). *Language and spatial cognition*. Studies in Natural Language Processing. Cambridge University Press.
- Johnson-Laird, P. N. (1977). Procedural semantics. *Cognition*, 5(3):189–214.
- Kemmerer, D. (1999). "Near" and "far" in language and perception. *Cognition*, 73(1):35–63.
- Kirby, S. (2002). The evolution of language. *Artificial Life*, 8:185–215.
- Levinson, S. C. (2003). *Space in Language and Cognition: Explorations in Cognitive Diversity*. Cambridge University Press.
- Levinson, S. C. and Wilkins, D. (2006). *Grammars of Space*. Cambridge University Press.
- Mainwaring, S., Tversky, B., Ohgishi, M., and Schiano, D. (2003). Descriptions of simple spatial scenes in English and Japanese. *Spatial Cognition and Computation*, 3(1):3–42.
- Rosch, E. (1975). Cognitive representations of semantic categories. *Journal of Experimental Psychology: General*, 140:192–233.
- Smith, K., Kirby, S., and Brighton, H. (2003). Iterated learning: A framework for the emergence of language. *Artificial Life*, 9(4):371–386.
- Spranger, M. (2008). World models for grounded language games. German diplom thesis, Humboldt-Universität zu Berlin.
- Spranger, M. (2011a). A Basic Emergent Grammar for Space. In Steels, L., editor, *Experiments in Cultural Language Evolution*. John Benjamins, Amsterdam.
- Spranger, M. (2011b). The Co-Evolution of Basic Spatial Terms and Categories. In Steels, L., editor, *Experiments in Cultural Language Evolution*. John Benjamins, Amsterdam.
- Spranger, M., Loetzsch, M., and Pauw, S. (2010). Open-ended grounded semantics. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, pages 929–934. IOS Press.
- Steels, L. (1995). A self-organizing spatial vocabulary. *Artificial Life*, 2(3):319–332.
- Steels, L. (2007). The Recruitment Theory of Language Origins. In *The Emergence of Communication and Language*, pages 129–151. Springer.
- Steels, L. (2011). Self-organization and Selection in Cultural Language Evolution. In Steels, L., editor, *Experiments in Cultural Language Evolution*. John Benjamins.
- Steels, L. and De Beule, J. (2006). Unify and merge in Fluid Construction Grammar. In *Symbol Grounding and Beyond: Proceedings of EELC, LNAI (4211)*, pages 197–223. Springer.
- Tenbrink, T. (2007). *Space, time, and the use of language: An investigation of relationships*. Walter de Gruyter.
- van Trijp, R. (2010). Strategy competition in the evolution of pronouns: A case-study of Spanish *léismo*, *laísmo* and *loísmo*. In *The Evolution of Language (EVOLANG 8)*, pages 336–343. World Scientific.

# An update on the minimal cell project: From the physics of solute encapsulation to the experimental modeling of cell communities.

Pasquale Stano<sup>1</sup>, Paolo Carrara<sup>1</sup>, Tereza Pereira de Souza<sup>2</sup>, and Pier Luigi Luisi<sup>1</sup>

<sup>1</sup>Biology Department, University of Roma Tre, Rome, Italy

<sup>2</sup>Pharmacy Institute, Friedrich Schiller University, Jena, Germany  
stano@uniroma3.it

## Abstract

The minimal cell (MC) project aims at understanding the emergence of cellular life by constructing experimental models of cells, according to a synthetic (constructive) biology approach. Our strategy – also known as the semi-synthetic one – is based on the encapsulation of the minimal number of biomolecular components inside lipid vesicles (liposomes). Being interested in studying the key step for constructing semi-synthetic cells, namely the physical entrapment of the solutes, we have recently reported that the mechanism of vesicle formation can lead to a spontaneous local increase in concentration of proteins inside vesicles (Luisi et al., ChemBioChem 2010, 11, 1989-1992). In particular, it was shown that the protein ferritin can reach intravesicle concentration of at least one order of magnitude higher when compared to the bulk (external) concentration. This self-organization phenomenon might give a rational account for the formation of functional cells from diluted solutions, and therefore help to understand the origin of metabolism. The effective encapsulation of solutes, however, is only one of the ways for achieving functional cells. The second route is fusion of vesicles or the exchange of solutes among vesicles (Caschera et al., J. Coll. Inter. Sci. 2010, 345, 561-565). Both processes allow the combination of different solutes to give compartments that can exhibit improved reactivity. Aiming at developing a realistic model for cooperative interactions among vesicles, we have recently developed a cell colony model. This is based on the formation of lipid vesicles clusters adherent to a solid substrate, representing a minimal model of cell communities. Here we summarize the most significant aspects of our recent activities.

## The physics of solute encapsulation

Looking at the physico-chemical mechanisms that have lead to the origin of cellular life, a still open question is whether functional cells have been originated from the encapsulation of an already developed metabolism (metabolism- or replicator-first scenarios), or whether the cell metabolism was entirely (or almost entirely) developed inside compartments (compartment-first scenario). In both cases, there are some aspects that need clarification, as the low probability of co-entrapping all required molecules in the same compartment in the first hypothesis, or the lack of permeability control in the second hypothesis (Luisi et al., 2010).

In particular, although the encapsulation of solutes into liposomes is a well-established field, especially due to the

large amount of work done in the field of drug delivery, we still miss a complete view of the physics underlying this important mechanism. In fact, with a few exceptions (Sun and Chiu, 2005; Dominak and Keating, 2007; Lohse et al., 2008), all experimental studies deal with the *average* entrapment yield, and no attention has been given to the entrapment behavior at the level of single vesicles, also due to technical difficulties.

We have recently started a systematic study on the encapsulation of biopolymers into lipid vesicles. This study was inspired by our report on the protein expression inside 200 nm (diameter) vesicles, that suggested the possible deviation from the expected intravesicle solute distribution (Souza et al., 2009).

As a model system, we have used the protein ferritin, an iron-storage protein, consisting of a nucleus of electron dense ferrihydrite-like iron salts surrounded by 24 protein subunits. Ferritin can be directly visualized as single molecule by electronmicroscopy, so that it becomes possible to directly count the number of ferritin molecules inside vesicles imaged via cryo-transmission electron microscopy.

After analyzing about 7,700 submicrometric vesicles (Fig. 1a), prepared by varying the concentration of ferritin, the preparation method and the membrane lipid composition, we have concluded that the encapsulation of this solute inside lipid vesicles does not follow the expected behavior. In our experimental conditions, this is given by the Poisson distribution of  $N$  solutes inside vesicles that are expected to entrap, on average,  $\mu$  solutes:

$$f(N) = e^{-\mu} \frac{\mu^N}{N!}$$

where  $f(N)$  represents the fraction of vesicles containing  $N$  ferritins, and  $\mu$  is the average expected number of ferritin molecules. The  $\mu$  value can be calculated from the vesicle volume  $V$ , and the ferritin concentration  $C$ :

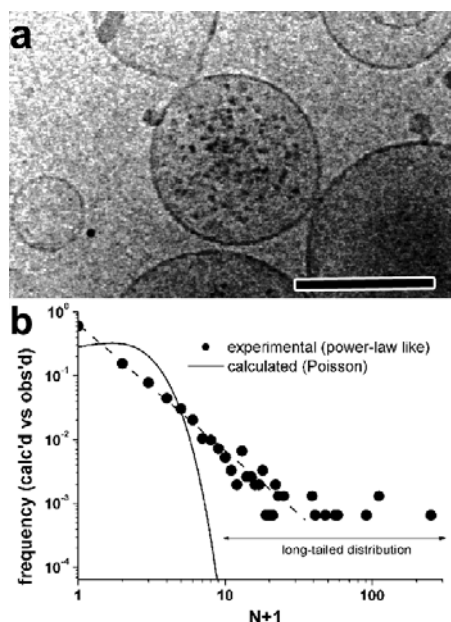
$$\mu = N_A C \cdot V$$

( $N_A$  being the Avogadro's number).

In particular, we have found that the distribution of ferritin-containing vesicles follows a power-law-like shape, characterized by an abnormally high amount of empty vesicles ( $N = 0$ ), a decreasing pattern at intermediate  $N$ , and – significantly – a non-zero long tail (Fig. 1b), which represents



the non negligible probability of co-entrapping a relatively high number of ferritins (up to hundreds), whereas the average expected value is of few units.



**Figure 1.** Entrapment of ferritin inside lipid vesicles. (a) Cryo-TEM electronmicrograph of a ferritin containing vesicle (size bar 200 nm). (b) Comparison between calculated Poisson ( $C = 4 \mu\text{M}$ , diam. 100 nm) distribution and experimental data profile. Redrawn after Luisi et al. (2010). The “long tail” feature has been highlighted. Note the logarithmic axes and the abscissa values shift ( $N+1$ ).

According to these results it appears that the co-encapsulation of several molecules in the same compartments is a physically possible process, and we believe that these observations contribute significantly for understanding the emergence of complex primitive cells from separated components. In fact, our results demonstrate that it is possible to form a solute-rich compartment even starting from diluted solution. This also implies that sluggishly reacting (diluted) systems might become reactive thanks to the spontaneous concentration increase inside lipid vesicles. Further studies about the mechanism will clarify our work hypothesis, based on weak and cooperative solute/membrane interactions, which affect the mechanism of vesicle closure (i.e., a process under kinetic control).

## Experimental models of cell communities

As we have anticipated, the co-entrapping of diverse solutes in the same compartment is not the unique process that can lead to solute-rich compartments starting from simpler ones. A complementary way is represented by all those mechanisms that have as a result the sharing of solutes among several compartments, in particular fusion and solute exchange. We have recently reported a study on the fusion between cationic and anionic vesicles as a way for reaching higher complexity, and loosely resembling the idea of symbiogenesis. In

particular, it was shown that oppositely charged vesicles can react (up to  $\sim 20\%$  yield) to neutralize their net charge and give rise to neutral species derived from the fusion of the vesicles (Caschera et al., 2010). As a consequence, the internal solutes, initially present in two vesicle populations, become co-encapsulated in the resulting new vesicles. We have reasoned that such fusion process, as well as the possibility of exchanging solutes among vesicles, could occur not only in suspended vesicles, but also in the case of vesicles forming small solid-supported communities. Here, the physical proximity of vesicles could not only favor such dynamical transformation, but simultaneously stabilize the community thanks to multiple physical interactions. Research is currently going on in our laboratory aimed at characterizing vesicle colonies with respect to their reproducible formation, physico-chemical stability, fusion, solute exchange as well as solute capture from the environment, and stability against flow (Carrara, 2010).

Thanks to this new experimental model we aim at studying the new “dimension” of *cell communities*, which is generally missing in the discussion on the origin of cellular life. Moreover, the model will allow a more direct investigation of communication between synthetic cells through the synthesis, release, uptake, and processing of diffusible species. This represents a concrete example of chemical communication, with possible implication in chemically-based information and communication technologies (ICTs).

A first attempt to use lipid vesicles for establishing a communication between synthetic and natural cells has been reported by Gardner et al. (2009).

**Acknowledgments.** We thank the following funding or networking agencies: SYNTHCELLS project (EU-FP6 043359), HFSP (RGP0033/2007-C), ASI (I/015/07/0), PRIN2008 (2008 FY7RJ4), SynBioNT, and COST Systems Chemistry action (CM0703). Dr. Frank Steiniger and Prof. Alfred Fahr (Jena) are acknowledged for cryoTEM analysis.

## References

- Carrara, P. (2010). *Constructing a Minimal Cell*. Ph.D. thesis, Department of Biology, Uniroma3 University.
- Caschera, F.; Stano, P.; Luisi, P. L. (2010). Reactivity and fusion between cationic vesicles and fatty acid anionic vesicles. *J. Coll. Inter. Sci.*, 345:561-565.
- Dominak, L. M.; Keating, C. D. (2007). Polymer encapsulation within giant lipid vesicles. *Langmuir*, 23:7148-7154.
- Gardner, P. M.; Winzer, K.; Davis, B. G. (2009). Sugar synthesis in a protocellular model leads to a cell signaling response in bacteria. *Nature Chem.*, 1:377-383.
- Lohse, B.; Bolinge, P.-Y.; Stamou, D. (2008). Encapsulation efficiency measured on single small unilamellar vesicles. *J. Am. Chem. Soc.*, 130:14372-14373.
- Luisi, P. L.; Allegretti, M.; Souza, T.; Steiniger, F.; Fahr, A.; Stano, P. (2010). Spontaneous protein crowding in liposomes: A new vista for the origin of cellular metabolism. *ChemBioChem*, 11:1989-1992.
- Souza, T.; Stano, P.; Luisi, P. L. (2009). The minimal size of liposome-based model cells brings about a remarkably enhanced entrapment and protein synthesis. *ChemBioChem*, 10:1056-1063.
- Sun, B.; Chiu, D. (2005). Determination of the encapsulation efficiency of individual vesicles using single-vesicle photolysis and confocal single-molecule detection. *Anal. Chem.*, 77:2770-2776.



# Reflecting on Open-Ended Evolution

Susan Stepney<sup>1</sup> and Tim Hovord<sup>1</sup>

<sup>1</sup>YCCSA, University of York, YO10 5DD, UK

susan@cs.york.ac.uk

## Abstract

We describe a computationally reflective object-oriented architecture suitable for incorporating open-ended innovation and emergent entities into simulations. This allows emergent properties to be reified into objects. This requires modifying the model, and the metamodel, by incorporating novel classes and metaclasses dynamically. The classes and metaclasses are modified by including them in the model through reflection. We argue that such computationally reflective introduction of novelty is necessary for true open-ended simulations.

## Introduction

Open-ended dynamics, supporting constant novelty generation, is a goal of ALife simulation.

Open-ended evolution has been defined as “a process in which there is the possibility for an indefinite increase in complexity” ([20], which also contains a comprehensive review of the concept in biology). Bedau [2] talks in terms of systems that exhibit “supple adaptation”, which involves them “responding appropriately in an indefinite variety of ways to an unpredictable variety of contingencies”. Open-ended novelty generation and evolution are features of biological life, but are proving hard to achieve in silico.

Classical evolutionary algorithms, with their fixed genome representations, can produce new things only within that limited representation. Evo-devo algorithms break out of this limitation, by allowing a genome to develop into a phenotype, but they are still confined to a single (albeit much richer) representation.

The desired continual increase in complexity is not merely a constant supply of new things (variations of a theme), or even of new kinds of things (speciation), but of new kinds of new kinds of things (major transitions, radical novelty, novel concepts). In computational terms, we might say we need a constant supply of new objects (the new things), new classes (new kinds of things, new representations), and new metaclasses (new kinds of kinds of things, new kinds of representations).

Here we take a computational modelling view of the problem, and describe what we believe are minimal requirements

for true open-ended dynamics in simulations: simulations that can modify their own model and metamodel as they execute. This implies that they can modify how they modify themselves. One key step on this route is the need to reify (“make concrete, or real”) emergent properties, as these are a rich source of novel concepts outside the language of the pre-existing system.

The structure of the rest of the paper is as follows. First we discuss the process of reifying emergent properties, both at the class and metaclass levels. Then we describe how a computational system can modify its own model and metamodel at runtime. Finally we specify a bootstrap architecture for such a self-modifying system.

## Extension, Intension, and Emergence

Consider an agent-based flocking simulation, implemented in some object-oriented (OO) programming language. A collection of boid objects exhibits various behaviours, and potentially forms flocks.

Assume the individual boid objects have names, eg Tweety, Cheeky, Polly, and ages, eg juvenile, adult, old. We can define particular sets of boids in two ways. An *extensional* definition explicitly enumerates the members:  $A = \{\text{Tweety}, \text{Polly}\}$ . An *intensional* definition is an implicit definition of membership in terms of properties of the members:  $B = \{b : \text{Boid} \mid b \text{ is juvenile}\}$ .

In an atemporal world of pure logic, a property is eternally either true or false, so extensionally defined set  $A$  and intensionally defined set  $B$  are either equal or not equal (have precisely the same members, or do not), and the difference in definitional approach is logically unimportant<sup>1</sup>. However, when properties are a function of time (as with stateful objects), an intensionally defined membership need not be static (for example, the membership of  $B$  may change as boids age). Hence  $A$  may equal  $B$  at one time, but not at another. In such a case, we need to be clear about whether the

<sup>1</sup>Except for such paradoxical definitions as “the set of all sets that are not members of themselves”, and other issues underpinning the foundations of mathematics, but we are not addressing these issues.

extent or the intent is the relevant defining property of our set of interest (for example, are we interested in Tweety and Polly, and “juvenile” is just a convenient shorthand for denoting them at this moment; or are we interested in juveniles, who just happen to be Tweety and Polly at this moment).

In general, we are interested in intensional property-based definitions, in potentially-changing collections of things that have certain properties in common (such as “all the blue birds more than a year old”), rather than in explicit but arbitrary collections (such as  $\{\text{Tweety}, (\emptyset, \text{Rover}), 42\}$ ). And we are more interested in generic intensionally-defined concepts (“flock”), than in specific one-off extensional collections (“those birds over there”).

In an OO program, nevertheless, collection objects (instances of Dictionaries, Sets, Lists, etc) are almost always extensional: they are static collections of the actual objects. The intent of such sets is only implicit (not captured in the code, except maybe through invariants or contracts), and much coding effort goes into maintaining this intent (explicitly adding and removing objects from the otherwise-static collection). This intent-implementing code, with its property-checking component, can be encapsulated inside a class. For example, consider the set of “all instances of class X”. This is an intensional definition: the set will contain different elements at different times, as instances of class X are created and destroyed. So in Smalltalk-80 [7] the (class) method `allInstances` returns an extensional set of all the instances of the class at the time of the message-send. The set itself does not change as objects are created and destroyed: a new message needs to be sent to the class to find the current value. The implementation hides the details of how this set is constructed each time; logically it is equivalent to constructing the set by examining every object and testing for the defining property.

## Emergence as implicit intension

Now consider the OO boid simulation. We point to an area of the screen, and say, “the flock is those boids”. So at any given moment, a flock appears to be an extensionally-defined set of boids:  $flock = \{\text{Tweety}, \text{Cheeky}, \dots, \text{Polly}\}$ . However, unlike a true extensional definition, the membership of the flock set can and does change, as boids leave and enter. This demonstrates that the flock is ‘really’ intensionally defined:  $flock = \{b : \text{Boid} \mid b \text{ has property } f\}$ . We just do not know what the intensional property  $f$  is, in advance<sup>2</sup>. The flocking property is emergent.

In some sense a flock is a ‘thing’, but it is not an object in our simulation, and there is no Flock class with which to capture and hide the intent-preserving code that tracks this

<sup>2</sup>Additionally, the property is probably somewhat fuzzy. For example, consider what might be the minimum size of the set *flock*. One boid, even two boids, do not make a flock. It has no well-defined answer; a flock is a fuzzy concept. (See, for example, the description of the Sorites Paradox in [11].)

set as boids enter and leave the flock. (Of course, we could have defined such a class, but that would require us to know beforehand the emergent properties; we assume here that we have not.)

We need some way to add this class and its intensional definition to the model and simulation *as and when the property emerges*. First we discuss different degrees of intensionalisation, and then a method and architecture for modifying the simulation with novel emergent properties.

## Intensionalising Emergence

We reify a specific flock by capturing it as an extensional object in the simulation, for example, as an instance of some generic Collection class (*theFlock* = `collectionInstance( $b_1, \dots, b_n$ )`). We can define the concept of flock in a new class Flock that explicitly captures the emergent intensional property, and so intensionalise the flock: *theFlock* = `flockInstance( $b_1, \dots, b_n$ )`. We can intensionalise an emergent property in a simulation in the following three ways, yielding different dynamics in the resulting system.

### External Instrumentation

Ordinary agents might remain blind to the existence of the emergent: it has no direct effect on them. For example, boids in a simple flocking simulation react to other boids independent of whether they are in a flock. (That is, their behavioural *rules* are unchanged, although of course their resulting *behaviour* is sensitive to the existence of the flock.)

In a simulation, we might add a FlockRecogniser subsystem, including a class FlockTag whose instances tag the detected flocks, and merely provide statistics on the simulation’s behaviour. Such instances would have no effect on the individual boids’ behaviour, whether within or outside a flock.

### Internal Detection

External instrumentation is the least interesting kind of reification, as the emergent is explicitly visible only to external observers. Crutchfield [5] talks about “intrinsic emergence”, where there are internal observer processes that can “take advantage of the emergent patterns”.

The next level of reification includes internal detection, whereby ordinary agents notice the existence of the emergent, and change their behaviour based on it. For example, a more sophisticated flocking simulation could have boids modified to be able to sense and interact directly with flock objects, preferring to move closer to a flock than to boids not in a flock, say. The flock object exists in the simulation, but is merely a derived consequence of the boids’ behaviours: it has no active behaviour of its own, it merely influences the behaviour of other objects.

## Reification

With full reification of the emergent, ordinary agents notice the existence of the emergent, change their behaviour, and are also directly affected by it. The emergent becomes an intensional entity in its own right. Being a component of the emergent then stops being defined merely as an extensional property (happening to being in the correct location to be in the extension, say), and becomes something that is granted by the emergent entity (membership rules, say).

For example, a reified flock object in a simulation might actively prevent boids from entering or leaving the flock. It would then be acting as a kind of ‘membrane’ around the flock. (We are *not* suggesting this happens in real flocks. Here we are simply exploring the kinds of things that a simulation might react to the presence of an agent: we are interested in getting complex open-ended dynamics in the simulation, not in faithfully replicating such processes occur in the real world.) The reified emergent becomes available in the simulation to be a firm component in further (higher-level) emergent behaviour.

The effect of the reified emergent on its constituents could be considered to be a form of downward causation [3, 22]. Although such a concept is anathema to reductionists, it is an everyday notion to sociologists. Reification of some societal constructs changes membership properties (for example, citizenship) from extensional (happening to be located in the country) to intensional (having the conceptual property of being a citizen) in exactly this way.

## Intensionalising Emergence internal

We have discussed modifications to the simulation to achieve several kinds of intensionalisation, to capture emergent properties as explicit entities within the simulation. In this section we propose how to achieve this *dynamically within the simulation*, through the use of computational reflection [16].

## Models

When writing a program, it is good software engineering practice to write a model of the program. For an OO program, that model is often written in an OO modelling language such as UML, identifying the classes, associations, interactions, behaviours, and so on. This model provides the abstract language of the concepts to be implemented in code. Even if no such model is written explicitly, it is implicit in the structure and dynamics of the written and executing code.

For example a (very simplified) class model of an agent-based boid simulation might look like figure 1. This is a model of the implemented code. Emergent (unimplemented) properties do not appear in this kind of model.

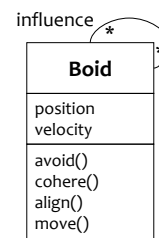


Figure 1: A (very simplified) UML class model of a boid simulation. There is a single Boid class, listing the attributes and methods.

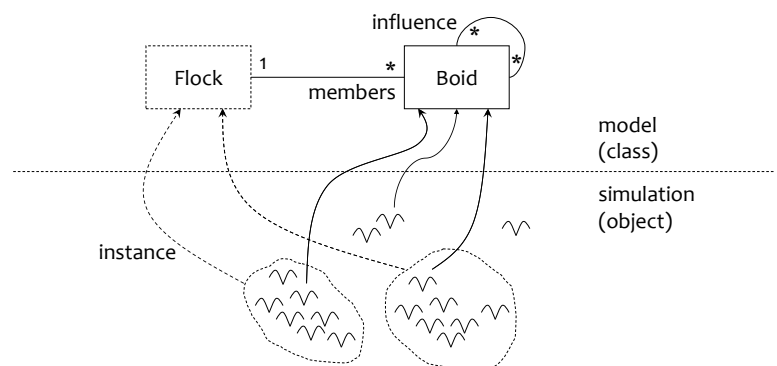


Figure 2: Model of Boids and emergent Flocks

## Emergent classes

Although the model of the simulation code does not include emergent concepts, we can build a (different) model that does. In this new model, the emergent is captured as an extensional object; it can then be intensionalised (its defining property captured in a class definition).

So we augment our model with an emergent class (which we draw as a dashed class box)<sup>3</sup>. This class captures the emergent property, and its instances. Figure 2 shows two levels: a model level with a normal class Boid and an emergent class Flock. We also show an object level view (a snapshot of the objects present during execution). The boid objects are instances of the Boid class. Some boid objects are members of flocks. We say that these emergent flock objects are instances of the emergent class Flock.

The emergent class might be a subclass of an existing ‘ordinary’ class in the model. For example, in an evolutionary system, a new kind of mutation operator might emerge ([8] discusses an example of an emergent macromutation, figure 3). In such a case we assume that the superclass is *abstract*, with neither intensional nor extensional instances of its own. On the other hand, the emergent class might be

<sup>3</sup>This is not part of UML, and so is an extension of the modelling language.

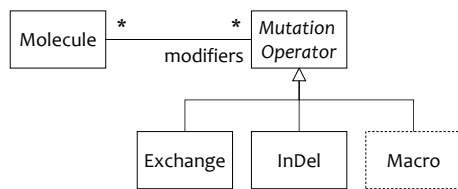


Figure 3: An example of an abstract superclass with many predefined subclasses

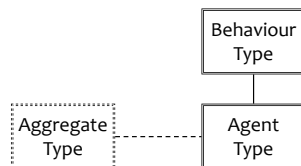


Figure 4: A (very simplified) metamodel of agent-based models. There is an Agent Type (an instance of which is the Boid class), and a Behaviour Type (instances include the boids' avoid and align behaviours). This metamodel has been augmented by an emergent Aggregate Type (an instance is the emergent Flock class).

a genuinely new kind of concept in the model, with no pre-existing superclass.

Once we have augmented our model with emergent objects and classes, we could build a new simulation with them as coded classes. But for an open-ended simulation, we need a system that can itself recognise such entities, and change its own model, *at run-time*, to include such intensionalised emergent classes dynamically.

## Metamodels

Changing the model (to allow for new kinds of executing objects), although necessary, is not sufficient for full open-endedness. We also need to change the metamodel, to allow new kinds of things in the model.

In an analogous way to how a model provides the language for writing the code, a metamodel provides the language for writing a *model*: it defines the kinds of things that can occur in the model (it is the model of the model). UML's metamodel includes concepts such as class and association. An agent-based modelling language metamodel would include concepts such as agent and behaviour. In the same way that models need to be augmented to include emergents, so do metamodels (figure 4).

Models and instances form a two-level modelling architecture. The Object Management Group (OMG) uses a four level modelling architecture [12, ch.8]: M0 = base instance (the objects in the simulation); M1 = model (defining the kinds of things in the simulation, such as Boid, Ant; written

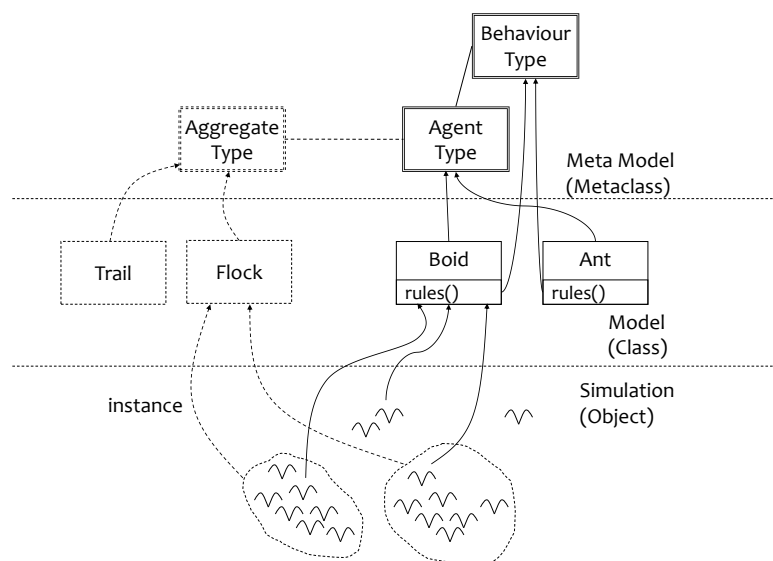


Figure 5: Metamodel and Model of an agent-based simulation

in, for example, UML); M2 = metamodel (defining an ontology, the kinds of thing in the model, eg Class and Association for UML models, AgentType for agent-based models (ABMs); also written in for example UML); and finally M3 = meta-metamodel (defining the kinds of thing in the meta-model, written in, for example, OMG's Meta Object Facility (MOF) language). Infinite regress is avoided by allowing the meta-metamodel to be written in MOF. Here we consider only the bottom three layers, M0-2.

Another example of this four level architecture is: M0 = executing program; M1 = a Python program; M2 = the Python programming language; M3 = BNF and denotational semantics. Changing the model is analogous to changing the *program*; changing the metamodel is analogous to changing the *programming language*.

## Emergent Metamodels

The metamodel of an ABM (figure 5) describes the kinds of things in an agent-based simulation: it has a metaclass AgentType. The emergent class like Flock in the model also needs a metaclass: it is an emergent AggregateType. So there can be emergent metaclasses too (where an emergent class is not an instance of some existing metaclass).

## Speciation and major transitions

We have seen three main kinds of reification:

1. Reifying an emergent subclass (for example, the macro-mutation class in figure 3). The concept already exists in the model (the superclass); the reified subclass is a variant of that concept.

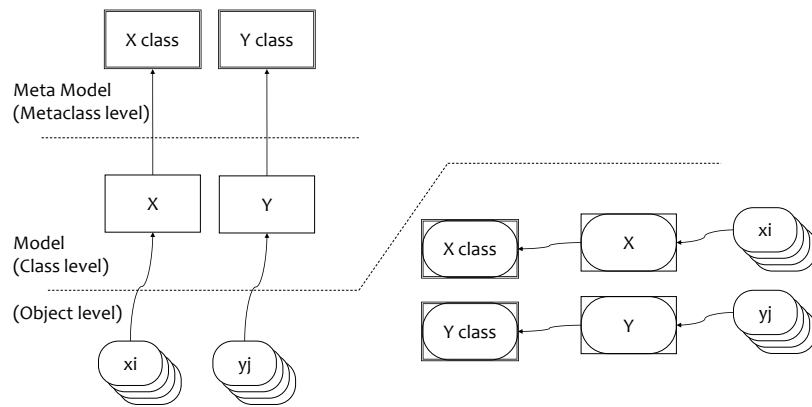


Figure 6: (left) the Smalltalk-80 Metaclass/Class/Object model as a three-layer model; (right) the implementation, all in the Object layer

2. Reifying an emergent class (for example, the Trail class in figure 5). The concept did not exist in the model (there is no relevant superclass), but does in the metamodel (once AggregateType is reified). The reified class is a new instance of that concept: the trail is a new kind of aggregate object, a new kind of thing with new kinds of behaviours, roughly analogous to a new species or genus in biology.
3. Reifying an emergent metaclass (for example, the aggregate type in figure 4). The concept did not exist in the metamodel: the aggregate type is a new kind of meta-object, a new concept in the language, roughly analogous to a major transition in evolutionary biology [17] (for example, the move from unicellular to multicellular organisms).

Such reification provides the requisite novelty generating power, when implemented in a computational system.

### Dynamic Models and Metamodels

The process of changing the model and metamodel needs to be dynamic, so that we can add reified emergent classes and metaclasses as they emerge and are recognised at run-time. Smalltalk-80 [7] provides an approach to this. Two fundamental concepts in Smalltalk-80 are: everything is an object; an object is an instance of some class. Since everything is an object, a class is an object, and so is an instance of some class, called its metaclass. So object *x* is an instance of class *X*, and class *X* is the (singleton) instance of its metaclass, referred to<sup>4</sup> as *X class*. Since everything is an object,

<sup>4</sup>In Smalltalk-80, metaclasses are not explicitly named. A metaclass can be referred to by sending the message `class` to the class's single instance. The value of this message expression is the metaclass. So the metaclass of class *X* can be referred to as *X class*. (Since there is also a class called `Class`, this terminology can lead to awkward constructions, such as "the class `Class class`".)

a metaclass is an object, and so is an instance of some class, the class `Metaclass`<sup>5</sup>.

So Smalltalk-80 has the objects, the classes (model) and metaclasses (metamodel) *all available as objects at runtime* (figure 6). All can be instantiated, deleted, and modified at runtime, via this *computational reflection* ("a reflective system is a computational system which is about itself in a causally connected way" [16]). Although Smalltalk-80 is not a pure reflective language, it does have reflective capabilities, and many others can be added programatically [6].

Other computationally reflective languages (ones that can modify themselves at run-time, to a greater or lesser extent) include Lisp, Prolog, Python, Ruby, and JavaScript.

### Examples of self-modifying and reflective systems

Suber [23] discusses self-amendment in the context of law making, and describes Nomic [10][23, appx.3], a (non-computer-based) law-based game where changing the rules (including the rule that players must obey the rules) is a move. Suber asks if it is possible either to make some rules unchangeable whilst preserving the power to amend others, or to irrevocably repeal the power to amend the rules.

Reflection is key in the branch of Artificial Intelligence concerned with "learning to learn", metamemory and metacognition [4, 14, 15, 18, 21, 24]. Learning changes the model; learning how to learn, learning a better learning algorithm, is changing the metamodel. Note that our concern here is not in high-level *cognition*, however, but in the role of reflection in open-ended *evolution*.

Biology is the ultimate self-modifying system. Hick-innbotham et al [9] describe a self-modifying computational

<sup>5</sup>Of course, since `Metaclass` is a class, it is the singleton instance of its metaclass, `Metaclass class`. And `Metaclass class` is a metaclass, so like all metaclasses, it is an instance of `Metaclass`. This circularity stops the potential infinite regress of needing metaclasses, etc. See [7, pp268-72] for details.

architecture inspired by biological DNA, RNA and protein machines. Tomita et al [25] use graph-rewriting automata with five kinds of rewriting rules, to implement self-replication. They discuss the possibility of embedding the graph rewriting program as a graph itself within the system, allowing for execution to modify which rules are applied. This is analogous to modifying the model at run-time; an analogy to modifying the metamodel would be to introduce new kinds of rewriting rules.

Reflection is proposed as the route to self-adaptive software systems [1, 16]. The architectural requirements specified in [1] differ from our own here, however, because the application domain is very different. For example, [1] is concerned with reflection on programming language concepts, subject to real world domain constraints; we are concerned with reflection on novelty generating mechanisms, and need to impose constraints in terms of some energy model (next). They are concerned with software engineering structuring, clear separation of model and metamodel layers and their respective concerns, and with performance; we are concerned with open ended novelty generation, and embrace the biologically-inspired ‘messiness’ of deliberately mixing layers of abstraction. Consequently, they carefully separate domain and reflective aspects, and keep the computation to do with reflection in the metamodel level only; our architecture of computation is orthogonal to the model and metamodel layers (next), to enable reflection at all levels, not only the metamodel reflecting on the model.

### An open-ended architecture

As discussed above, computational reflection provides a route to open-ended novelty. As Maes [16] says: “A language with reflective facilities is **open-ended**: reflection makes it possible to make (local) specialised interpreters of the language, from within the language itself.”

Reflection provides the computational mechanism, but we also need an architecture within which to generate and run the open-ended code. Here we describe an architecture for such a system. We use OO terminology; this specific paradigm, although well-suited, is not necessary for the architecture, just some analogue of the underlying concepts in a reflective programming language.

We define only a bootstrap architecture. The whole point of computational open-ended novelty generation is for the system to modify this architecture at run-time.

The key feature is that the three levels – instance, class, and metaclass – all exist as executing and modifiable objects in the system at run-time. For the bootstrap, we separate the system into three subsystems: an initial seed application, the observer-reifier-modifier (ORM) intentionaliser, and the virtual machine (VM). The seed application, for example, some agent-based simulation, acts as the raw material from which the open ended novelty grows. The other two subsystems are described below. See figure 7.

### ORM Intentionaliser: modifying the models

Our framework for intensionalising emergent structures has three components:

1. emergence **observers**, that observe novel emergent structures and behaviours
2. emergence **reifiers**, that intensionalise the recognised types, and add the relevant classes or metaclasses into the run-time, thereby changing the model or metamodel
3. model **modifiers**, that modify the simulation (instances, classes, or metaclasses) to exploit the reified structures

In [1], a distinction is made between structural reflection (reification of structural aspects such as data types) and behavioural reflection (reification of computations and their behaviours). It is crucial that emergence recognisers capture patterns both of structure and of behaviour: at different levels of emergence features can appear to be either ‘particles’ or ‘processes’ [22].

The ORM subsystem therefore includes *ObserverType*, *ReifierType*, and *ModifierType* metaclasses, and bootstrap class instances of these, to provide the meta-functionality. For example, we might have the class *Eye* as a bootstrap instance of *ObserverType*, whose own instances observe the simulation for particular spatial and temporal patterns that indicate emergence. An *Eye* instance might detect a flock- or trail-like emergent. It notifies a suitable *Reifier* instance, which can appropriately intensionalise the emergent, for example, as an internally detectable object. A suitable *Modifier* instance then modifies other classes in the simulation so that their instances can detect the new objects. It might also modify their behaviours to use the detected information, or, in an evolutionary simulation, allow these modified behaviours to evolve.

Key to the overall architecture is the fact that the simulator is reflective, not just at the core agent level, but throughout. Hence a bootstrap observer (for example) can observe not only novel agent patterns, but also novel observation, reification, and modification patterns, which can then be reified and modified appropriately. We bootstrap with *Hammer* and *Eye* classes; later *Spanner* and *Ear* classes can emerge and be reified. Eventually new *ModifierType* metaclasses could be reified. Hence the simulation can not only change itself, it can change the way it changes itself (this does imply requirements on the representation of modifier rules [14]).

Being able to modify the modifier, being able to produce new kinds of ways of recognising, reifying and modifying the simulation, closes the self-referential loop, and produces a truly open-ended system.

### Virtual machine: constraints

The virtual machine provides whatever run-time support is needed for the ORM architecture, in the usual manner (at

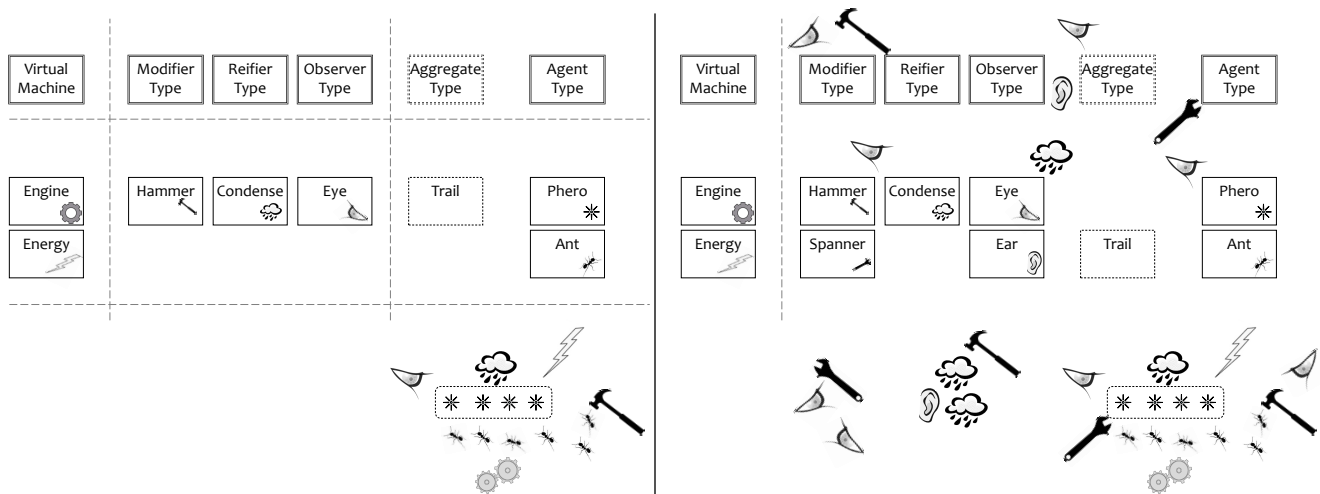


Figure 7: The architecture, showing metamodel, model (class boxes show an instance icon), and instance layers (horizontal dashed lines), and VM, ORM, and ABM subsystems (vertical dashed lines). (left) Minimal self-modification: the layers and subsystems are well-defined, the bootstrap ORM objects observe and reify the emergent trail class and modify the ABM objects, the ORM model and metamodel are fixed. (right) Constrained self-modification: the ORM components observe and modify the ABM and ORM objects, model, and metamodel (but not the VM), reifying emergent ORM components, and potentially modifying the kind of modifiers.

a minimum, compilation, dynamic object communication, and error handling). In addition, it provides some form of constraint on the modification processes. The research challenge is to achieve framework behaviour that allows a simulation to exploit emergent novelty without dissolving into chaos. A completely unconstrained framework could well modify itself out of existence. Some form of constraint, for example an analogue of conservation of energy, might be needed to allow the system to develop in interesting directions without devolving into a mess of object soup.

However, a completely constrained system, that allows no modification, no intensionalisation, is static and cannot achieve open-ended dynamics. This is the state of most classic ABM simulations.

It seems plausible that some degree of constraint between a totally static mode and meta model, and total freedom, is required; this is possibly some “edge of chaos” [13] requirement. Hence the role of the constraint is to help the system self-organise to maximally complex patterns of structure and behaviour.

## Modifying the VM

If the virtual machine is implemented in the same language and at the same level as ORM, it could potentially also be a target of the self-modification process. Here we assume that the constraint part is to be unmodifiable, for the reasons given above, but the interpreter or message handler part is a valid target of modification.

Consider a Smalltalk-80 implementation. The simulation and modifier objects are Smalltalk-80 objects, and are im-

plemented (given their execution semantics) in a Smalltalk-80 VM. A suitably defined physics engine could be included at the object level, and be subject to the same modification processes as the objects themselves.

## Discussion

Consideration of a metamodel of emergence has led to the insight that emergent properties are emergent intensional definitions. The difference exhibits itself in simulations, where the emergent properties are observed via instrumentation, rather than reified directly. If the emergent properties are reified and intensionalised, with their own definitions and behaviours, they can become the kind of agents that result in (further) emergent properties.

In order for these kinds of emergent innovation to be included in a simulation, the simulation needs to be able to modify its own model, and metamodel, dynamically (at run time). We contend that **for a simulation to exhibit open ended dynamics, it must include a form of computational reflection** that allows it to modify its own model and metamodel as the simulation is running.

We have specified the design of an open-ended architecture. (The next stage of work is to develop a prototype implementation.) This architecture has the instances, model, and metamodel all available for modification at run-time. It has three subsystems: a virtual machine providing run-time support and modification constraints, an observer-reifier-modifier intensionalizer, and a seed application. This is a *bootstrap* architecture: successful self-modification will modify this architecture.



Rosen [19, §10a] argues that the difference between an organism and a mechanism is that an organism “is closed to efficient causation”, and that a mechanism cannot be so closed. He uses Aristotle’s term “efficient cause” as the cause that brings something about. He argues that life is self-defining, self-causing, autopoietic; but that simulations cannot be, that simulations require something outside the system to define them. We claim that the reflective approach and bootstrap architecture described above *can* allow simulations to be similarly self-defining, self-generating, self-causal, and hence to exhibit some of the properties Rosen requires for life.

## Acknowledgments

Out thanks to Paul Andrews, Ed Clark, Tim Clarke, Simon Hickinbotham, Adam Nellis, Mungo Pay, Fiona Polack, Adam Sampson, Jon Timmis, Emma Uprichard, and Peter Young, for helpful discussions, and to the anonymous referees for their suggestions.

The work described here is part of the CoSMoS<sup>6</sup> project, funded by EPSRC grant EP/E053505/1 and a Microsoft Research Europe PhD studentship.

## References

- [1] Jesper Andersson, Rogerio de Lemos, Sam Malek, and Danny Weyns. Reflecting on self-adaptive software systems. In *SEAMS’09*, pages 38–47. IEEE, 2009.
- [2] Mark A. Bedau. The nature of life. In Margaret A. Boden, editor, *The Philosophy of Artificial Life*. Oxford University Press, 1996.
- [3] Donald T. Campbell. ‘Downward Causation’ in hierarchically organised biological systems. In Francisco Jose Ayala and Theodosius Dobzhansky, editors, *Studies in the Philosophy of Biology: reduction and related problems*, chapter 11, pages 179–186. Macmillan, 1974.
- [4] Michael T. Cox. Metacognition in computation: A selected research review. *Artificial Intelligence*, 169(2):104–141, 2005.
- [5] James P. Crutchfield. The calculi of emergence. *Physica D*, 75:11–54, 1994.
- [6] Brian Foote and Ralph E. Johnson. Reflective facilities in Smalltalk-80. In *OOPSLA’89*, pages 327–335. ACM Press, 1989.
- [7] Adele Goldberg and David Robson. *Smalltalk-80: The Language and its Implementation*. Addison-Wesley, 1983.
- [8] Simon Hickinbotham, Edward Clark, Susan Stepney, Tim Clarke, Adam Nellis, Mungo Pay, and Peter Young. Diversity from a monoculture: effects of mutation-on-copy in a string-based artificial chemistry. In *ALife XII*, pages 24–31. MIT Press, 2010.
- [9] Simon Hickinbotham, Susan Stepney, Adam Nellis, Tim Clarke, Edward Clark, Mungo Pay, and Peter Young. Embodied genomes and metaprogramming. In *ECAL 2011*. MIT Press, 2011.
- [10] Douglas Hofstadter. Metamagical themas. *Scientific American*, June 1982.
- [11] Dominic Hyde. Sorites paradox. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2008 edition, 2008. <http://plato.stanford.edu/archives/fall2008/entries/sorites-paradox/>.
- [12] Anneke Kleppe, Jos Warmer, and Wim Bast. *MDA Explained: the Model Driven Architecture: practice and promise*. Addison-Wesley, 2003.
- [13] Chris G. Langton. Computation at the edge of chaos: phase transitions and emergent computation. *Physica D*, 42:12–37, 1990.
- [14] Douglas B. Lenat and John Seely Brown. Why AM and EURISKO appear to work. *Artificial Intelligence*, 23:269–294, 1984.
- [15] Luís Seabra Lopes and Aneesh Chauhan. Open-ended category learning for language acquisition. *Connection Science*, 20(4):277–297, 2008.
- [16] Pattie Maes. Concepts and experiments in computational reflection. In *OOPSLA’87*, pages 147–155. ACM Press, 1987.
- [17] John Maynard Smith and Eörs Szathmáry. *The Major Transitions in Evolution*. Oxford University Press, 1995.
- [18] Thomas O. Nelson. Metamemory: A theoretical framework and new findings. *Psychology of Learning and Motivation*, 26:125–173, 1990.
- [19] Robert Rosen. *Life Itself*. Columbia University Press, 1991.
- [20] Kepa Ruiz-Mirazo, Jon Umerez, and Alvaro Moreno. Enabling conditions for open-ended evolution. *Biology and Philosophy*, 23:67–85, 2008.
- [21] Jürgen Schmidhuber, Jieyu Zhao, and Nicol N. Schraudolph. Reinforcement learning with self-modifying policies. In Thrun and Pratt [24], pages 293–309.
- [22] Susan Stepney, Fiona Polack, and Heather Turner. Engineering emergence. In *ICECCS 2006*, pages 89–97. IEEE, 2006.
- [23] Peter Suber. *The Paradox of Self-Amendment: a study of law, logic, omnipotence, and change*. Peter Lang, 1990. <http://www.earlham.edu/~peters/writing/psa/>.
- [24] Sebastian Thrun and Lorien Y. Pratt, editors. *Learning to Learn*. Kluwer, 1997.
- [25] Kohji Tomita, Satoshi Murata, and Haruhisa Kurokawa. Self-description for construction and computation on graph-rewriting automata. *Artificial Life*, 13(4):383–396, 2007.

<sup>6</sup><http://www.cosmos-research.org>

# Self-Organized Flocking with a Heterogeneous Mobile Robot Swarm

A. Stranieri<sup>1</sup>, E. Ferrante<sup>1</sup>, A. E. Turgut<sup>1</sup>, V. Trianni<sup>1</sup>, C. Pinciroli<sup>1</sup>, M. Birattari<sup>1</sup> and M. Dorigo<sup>1</sup>

<sup>1</sup>Université Libre de Bruxelles, 1050, Belgium

{astranie, eferrante, aturgut, vtriani, cpinciro, mbiro, mdorigo}@ulb.ac.be

## Abstract

In this paper, we study self-organized flocking in a swarm of behaviorally heterogeneous mobile robots: aligning and non-aligning robots. Aligning robots are capable of agreeing on a common heading direction with other neighboring aligning robots. Conversely, non-aligning robots lack this capability. Studying this type of heterogeneity in self-organized flocking is important as it can support the design of a swarm with minimal hardware requirements. Through systematic simulations, we show that a heterogeneous group of aligning and non-aligning robots can achieve good performance in flocking behavior. We further show that the performance is affected not only by the proportion of aligning robots, but also by the way they integrate information about their neighbors as well as the motion control employed by the robots.

## INTRODUCTION

Flocking is the cohesive and aligned motion of a group of individuals along a common direction. All studies about flocking within computer science and robotics root back to the seminal work of Reynolds (1987). He was the first to simulate flocking of birds based on three behaviors: *separation* — individuals try to keep a minimum distance between their neighbors, *cohesion* — individuals try to stay together with their neighbors, and *alignment* — individuals try to match their velocities to the average speed of their neighbors. The vast majority of the studies about flocking assume that all the robots in the swarm are behaviorally identical and exploit the three behaviors described above.

In this paper, we consider flocking in a behaviorally heterogeneous swarm of robots. All robots in the swarm use the separation and the cohesion behavioral rule. However, only a fraction of the robots, which we call the *aligning* robots, uses the alignment behavior. The rest of the robots, which we call the *non-aligning* robots, do not use the alignment behavior.

We believe that studying heterogeneity in alignment in self-organized flocking is very important from the practical point of view. The alignment behavior is more demanding in terms of robotics hardware requirements than the separation and cohesion behaviors. In fact, it requires either an elaborate sensing device, through which robots can detect the

orientation of neighboring robots or, as explained in this paper, a communication device. Therefore, understanding if a swarm can achieve flocking with only a few aligning robots can support the design of swarms with minimal hardware requirements.

We conduct simulation-based experiments and we measure self-organized flocking performance in terms of the degree of group order, group cohesiveness and average group speed. With respect to these criteria, we found that the swarm achieves good flocking performance when the proportion of aligning robots is high. Conversely, this performance decreases as the proportion gets lower. To tackle this problem, we propose a new model of robot motion. In the new model, non-aligning robots modulate their forward speed, instead of moving at a fixed forward speed as the other robots.

The rest of the paper is organized as follows. In the next section, we present the related works in flocking, starting from studies in biology and then in robotics. We then introduce our heterogeneous flocking model, the robots and we explain how we implement flocking on the robots. Subsequently, we describe the experimental setup, the metrics and the results. Finally, we conclude the paper and propose future directions of research.

## RELATED WORK

Flocking is a widely observed phenomenon in social animals (Camazine et al., 2001) such as locusts (Buhl et al., 2006), birds (Ballerini et al., 2007) or human beings (Dyer et al., 2008). Animal groups show a great diversity in their population due to the differences in age, morphology (Krause et al., 1998), nutritional state (Krause, 1993), personality (Michelena et al., 2010), and leadership status (Reebs, 2000) of the individuals. This diversity mainly results in behavioral differences among the individuals. Couzin et al. (2002) showed that behavioral differences between the individuals in a group change both the dynamics and the organization of the group. Subsequently, Couzin et al. (2005) conducted a seminal study about leadership in animal groups. They modeled a heterogeneous group of in-

dividuals of which only a few are aware of a target direction. They showed that the few informed individuals are able to move the whole group along the target direction. In Janson et al. (2005), the authors propose a model to explain how scout bees are able to direct large swarms of uninformed bees towards a new nesting site. Even when the proportion of scout bees is low, they are able to lead the swarm by flying through it at a slightly faster speed. Sayama (2009) presented the preliminary results obtained in simulation using the Swarm Chemistry framework. They studied the movement of a swarm consisting of two different chemical species, and found that a chaser-escaper relationship between the two different populations of agents is established. More recently, Diwold et al. (2011) showed how a swarm can still fly towards a common direction even when the agents are not all aligned, and when the location of the nesting site is not known with precision.

In robotics, most of the studies about flocking assume a homogeneous set of behaviorally equivalent individuals. One of the earliest studies in robotics was performed by Mataric (1994). She devised a set of “basis behaviors” to implement flocking in a group of robots: safe-wandering, aggregation, dispersion and homing. With the proposed set of behaviors, robots are able to move cohesively towards a homing direction. Kelly and Keating (1996), following a behavior-based approach, designed a leader-following behavior to realize flocking. Hayes and Dormiani-Tabatabaei (2002) proposed a flocking behavior having collision avoidance and alignment behaviors based on local range and bearing measurements. Spears et al. (2004) proposed a framework based on artificial physics. The robots were able to form a regular lattice structure using attraction/repulsion virtual forces and move along a direction indicated by a light source in the environment. Holland et al. (2005) proposed a flocking behavior for unmanned ground vehicles based on separation, cohesion and alignment behaviors. Turgut et al. (2008) proposed a flocking behavior based on separation/cohesion and alignment behaviors. They implemented this behavior in robots with limited sensing capabilities and conducted a systematic study on the effect of sensing noise in heading measurement on flocking. In a recent study, Moslinger et al. (2009) proposed a flocking behavior for robots with limited sensing capabilities. It is based on only attraction and repulsion behaviors. By adjusting the sizes of attraction and repulsion zones, they achieved flocking for a small group in a constrained environment.

Other works in robotics considered a group of behaviorally heterogeneous robots. Momen et al. (2007) studied flocking with a heterogeneous robotic swarm inspired by mixed-species foraging flocks of birds (Graves and Gotelli, 1993). Using simulations, they showed some aspects of mixed-species flocking, such as behavioral differences in their attraction and repulsion rules. Çelikkanat and Şahin (2010), inspired by Couzin et al. (2005) extended the flock-

ing behavior proposed by Turgut et al. (2008) and created a heterogeneous robot swarm by informing some of the robots about a target direction. Recently, in another follow-up study, Ferrante et al. (2010) introduced a new communication strategy to improve flocking performance in case of both static and changing target directions.

To the best of our knowledge, most of the studies in swarm robotics about self-organized flocking have not considered diversity in alignment capabilities.

## METHOD

We follow a design method based on the artificial physics framework introduced by Spears et al. (2004). According to this method, robots exert virtual forces on each other. The swarm consists of aligning and non-aligning robots. Aligning robots are subject to the following virtual forces

$$\mathbf{f} = \alpha_1 \mathbf{p} + \beta_1 \mathbf{h},$$

whereas for the non-aligning robots the virtual force is computed as

$$\mathbf{f} = \alpha_2 \mathbf{p}.$$

We define  $\mathbf{p}$  as the *proximal control vector* and  $\mathbf{h}$  as the *alignment control vector*. The proximal control vector  $\mathbf{p}$  accounts for attraction and repulsion rules for keeping the robot together with its neighbors and to avoid collisions.

The alignment control vector  $\mathbf{h}$  is used to make the aligning robots match the average heading direction of its neighboring aligning robots. The parameters  $\alpha_1$ ,  $\beta_1$  and  $\alpha_2$  are used to adjust the contribution of the corresponding vectors.

### Proximal control

Let  $m_p$  denote the number of neighbors of a robot within a range  $D_p$ . Let also  $d_i$  and  $\phi_i$  denote the relative range and bearing of the  $i^{th}$  neighbor, respectively. The proximal control vector  $\mathbf{p}$  is given by:

$$\mathbf{p} = \sum_{i=1}^{m_p} p_i(d_i) e^{j\phi_i}.$$

$p_i$  is calculated as a function of  $d_i$  using a force function derived from the Lennard-Jones potential function, which results in the formation of regular structures as shown in Hettiarachchi and Spears (2009):

$$p_i(d_i) = 12\epsilon \left[ \frac{d_{des}^{12}}{d_i^{13}} - \frac{d_{des}^6}{d_i^7} \right].$$

The parameter  $\epsilon$  determines the strength of the attractive and repulsive force, and  $d_{des}$  is the desired distance between the robots.

## Alignment control

Let  $\theta_0$  denote the orientation of a given robot. Furthermore, let  $m_a$  denote the number of aligning robots within the range  $D_a$  of this robot, and  $\theta_i, i \in \{1, \dots, m_a\}$  their orientation. All orientations are expressed in the body-fixed reference frame of the robot under consideration<sup>1</sup>. The robot calculates the alignment control vector, that is, the average orientation of the  $m_a$  robots, including its own:

$$\mathbf{h} = \frac{\sum_{i=0}^{m_a} e^{j\theta_i}}{\|\sum_{i=0}^{m_a} e^{j\theta_i}\|},$$

where  $\|\cdot\|$  denotes the norm of a vector.

## Motion control

We present two motion control rules. The two rules differ in the way the forward speed  $u$  and the angular speed  $\omega$  are determined. The first rule is denoted as *constant forward speed motion control* (henceforth CMC). In CMC, robots are always moving at a constant forward speed, but can change their angular speed. According to the second rule, denoted as *variable forward speed motion control* (henceforth VMC), robots move not only at a variable angular speed but also at a variable forward speed.

**CMC:** The forward speed is kept constant at

$$u = U.$$

The angular speed is proportional to the angular component of the total force  $\mathbf{f}$ . Hence, it ignores the magnitude  $\|\mathbf{f}\|$  of the force:

$$\omega = K\angle\mathbf{f}.$$

**VMC:** First, let  $\mathbf{f}_x = \|\mathbf{f}\| \cos(\angle\mathbf{f})$  and  $\mathbf{f}_y = \|\mathbf{f}\| \sin(\angle\mathbf{f})$  denote the projection of the total force  $\mathbf{f}$  on the  $x$ -axis and  $y$ -axis of the robot body-fixed reference frame respectively. Accordingly, the forward speed  $u$  is directly proportional to the  $x$  component of the total force and the angular speed  $\omega$  is directly proportional to the  $y$  component of the force. Hence:

$$\begin{aligned} u &= K_1 \mathbf{f}_x \\ \omega &= K_2 \mathbf{f}_y. \end{aligned}$$

$K, K_1, K_2$  are constants, whose values are given in Table 1.

In this work, we consider and study two different cases in which we vary the motion control rule applied to the non-aligning robots. In the first case, referred as the *CMC-CMC*

<sup>1</sup>In our study, we define two reference frames, both of which use the right-hand convention. One is the reference frame common to all of the robots, which is available due to the light source. The other is the body-fixed reference frame specific to each robot. The body-fixed reference frame is fixed to the center of a robot: its  $x$ -axis points to the front of the robot and its  $y$ -axis is coincident with the rotation axis of the wheels.

case, all robots share the same motion control rule, that is, CMC. In the second case, referred as the *CMC-VMC* case, aligning robots use CMC, whereas non-aligning robots use VMC.

## FLOCKING WITH ROBOTS

In this study, the swarm is composed of simulated versions of the foot-bot robot developed by Bonani et al. (2010). The foot-bot is a differentially-driven mobile robot with the following sensors and actuators: i) A light sensor used to measure the orientation of robot ( $\theta_0$ ) with respect to a light source present in the environment perceived by all robots. ii) A range and bearing sensing and communication device (henceforth called RAB), with which a robot can communicate with its neighbors and perceive their range and bearing measurements (Roberts et al., 2009). iii) Two wheels actuators, that are used to control independently the left and right wheels speed of the robot.

To achieve proximal control with the foot-bot the RAB is used for measuring the relative range and bearing  $d_i$  and  $\phi_i$  of the  $i_{th}$  neighbor. For achieving alignment control, we use communication to simulate orientation sensing as in Turgut et al. (2008). In particular, each aligning robot sends its orientation, expressed in the global reference frame, using the communication unit present in the RAB. At the same time, it receives the orientation  $\theta_i$  of its  $i_{th}$  neighboring aligning robot. It transforms this angle into its body-fixed reference frame. In this way, we are able to simulate a robot sensing the orientation of its neighboring aligning robots.

To achieve motion control, we first limit the forward speed within  $[0, U_{max}]$ , and the angular speed within  $[-\Omega_{max}, \Omega_{max}]$ . We then use the differential drive model used in Turgut et al. (2008) to convert the forward speed  $u$  and the angular speed  $\omega$  into the linear speeds of the left ( $N_L$ ) and right ( $N_R$ ) wheel:

$$N_L = \left(u + \frac{\omega}{2}l\right),$$

$$N_R = \left(u - \frac{\omega}{2}l\right),$$

where  $l$  is the distance between the wheels.

The values of the constants that we used in our experiments are given in Table 1.

## EXPERIMENTS

We execute simulation-based experiments with a swarm of foot-bots using the ARGOS simulator (Pinciroli et al., 2011), an open-source<sup>2</sup>, plug-in based, multi-physics engine simulator.

<sup>2</sup><http://iridia.ulb.ac.be/argos/>

Variable	Description	Value(s) / Range
$N$	Number of robots	$\{25, 100\}$
$\rho$	Prop. of aligning robots	$\{0.4, 0.8\}$
$\beta_1/\alpha_1$	Align. robots parameters	$\{1, 2, 4, 6, 8, 10\}$
$\alpha_2$	Non align. robots parameter	$\{1, 2, 4, 6, 8, 10\}$
$U$	Maximum forward speed	1.5 cm/s
$K$	CMC angular gain	0.5 1/s
$K_1$	VMC linear gain	0.25 s/kg
$K_2$	VMC angular gain	0.1 s/(kg · m)
$l$	Inter-wheel distance	0.1 m
$U_{max}$	VMC max forward speed	20 cm/s
$\Omega_{max}$	VMC max angular speed	$\pi/2$ rad/s
$\epsilon$	Strength of pot. function	0.5
$d_{des}$	Inter-robot distance	0.6 m
$\sigma$	Amount of noise	0.1
$T$	Experiment duration	600 secs

Table 1: Experimental values or range of values for all constants and variables

## Experimental setup

At the beginning of each experiment,  $N$  mobile robots are randomly placed (position and orientation-wise) with a proportion  $\rho \in [0, 1]$  of aligning robots. The density of robots is kept fixed and equal to 6 robots per square meter on a square shaped area. A light source is placed at a fixed position in the environment, far away from the swarm, to provide the common reference frame.

In the experiments, noise is added to the orientation measurement and the angle of the proximal control vector. Noise is modeled as a uniformly distributed random variable within the range  $[-\sigma\pi, \sigma\pi]$ .

We conduct experiments considering the two different cases of motion control.

**CMC-CMC** In this case, all robots use CMC. Here, we study the effect of the ratio  $\frac{\beta_1}{\alpha_1}$ , and we do not change  $\alpha_1$  and  $\beta_1$  independently, since CMC does not utilize the magnitude of  $\mathbf{f}$ , but only its angular component. As such, multiplying both  $\alpha_1$  and  $\beta_1$  with the same constant value will produce no difference in the robot motion. For the same reason,  $\alpha_2$  does not effect the robot motion.

**CMC-VMC** In this case, aligning robots use CMC whereas non-aligning robots are using VMC. For the non-aligning robots, the magnitude of  $\mathbf{f}$  plays a role in their motion. Thus, additionally to the effect of changing  $\frac{\beta_1}{\alpha_1}$  of the aligning robots, we study the effect of changing  $\alpha_2$  of the non-aligning robots.

We show the results in heterogeneous self-organized flocking with medium ( $N = 25$ ) and large ( $N = 100$ ) swarm sizes and with low ( $\rho = 0.4$ ) and high ( $\rho = 0.8$ ) proportions

of aligning robots. We study the effect of changing the ratio  $\frac{\beta_1}{\alpha_1} \in \{1, 2, 4, 6, 8, 10\}$  and, for the heterogeneous case, we also study the effect of changing  $\alpha_2 \in \{1, 2, 4, 6, 8, 10\}$ , but we report here only the results obtained with the best case, that is,  $\alpha_2 = 10$  (refer to Stranieri et al. (2011) for the complete set of results ). In our supplementary page (Stranieri et al., 2011), we also report the flocking performance as a function of  $\rho \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ .

For each experimental setting, we execute  $R$  runs and report median and interquartile range of the results. The duration of one run is  $T$  simulated seconds.

We study how the heterogeneous flocking performance is influenced by: i) the way robots implement their motion (CMC-CMC motion versus CMC-VMC motion), ii) the parameters that affect the strength of the proximal control vector and of the alignment control vector, that is,  $\frac{\beta_1}{\alpha_1}$  and  $\alpha_2$ , and iii) the ratio of aligning robots  $\rho$ .

We also experiments in the VMC-VMC case, but we didn't obtain any positive results, even with  $\rho = 1$ .

## Metrics

In this study, we are interested in having a swarm of robots that move cohesively as a single group. Furthermore, the swarm should be aligned towards the same direction and move towards it as fast as possible. We use three metrics to measure the degree of attainment of these objectives: order, group cohesion and rescaled group speed.

**Order:** The order metric  $\psi$  measures the angular order of the robots (Vicsek et al., 1995),  $\psi \approx 1$  when the group shares a common heading and  $\psi \ll 1$  when each robot is pointing in a different direction. The order is defined as:

$$\psi = \frac{1}{N} \left\| \sum_{i=1}^N e^{j\theta_i} \right\|.$$

**Group cohesion:** To measure group cohesion  $\xi$ , we determine the number of groups  $g$  present at the end of each experiment (Couzin et al., 2005). Group cohesion is computed as:

$$\xi = 2 - \min(2, g).$$

and therefore takes values in  $\{0, 1\}$ .

**Rescaled Group speed:** We calculate the average group speed as:

$$s = \left\| \frac{\mathbf{c}_T - \mathbf{c}_0}{T} \right\|,$$

where  $\mathbf{c}_T$  and  $\mathbf{c}_0$  are the position of the center of mass of the swarm at the end and at the beginning of the experiment, respectively. We then rescale the average group speed:

$$s_r = \frac{s}{U},$$

where  $U$  is the maximum forward speed of CMC.

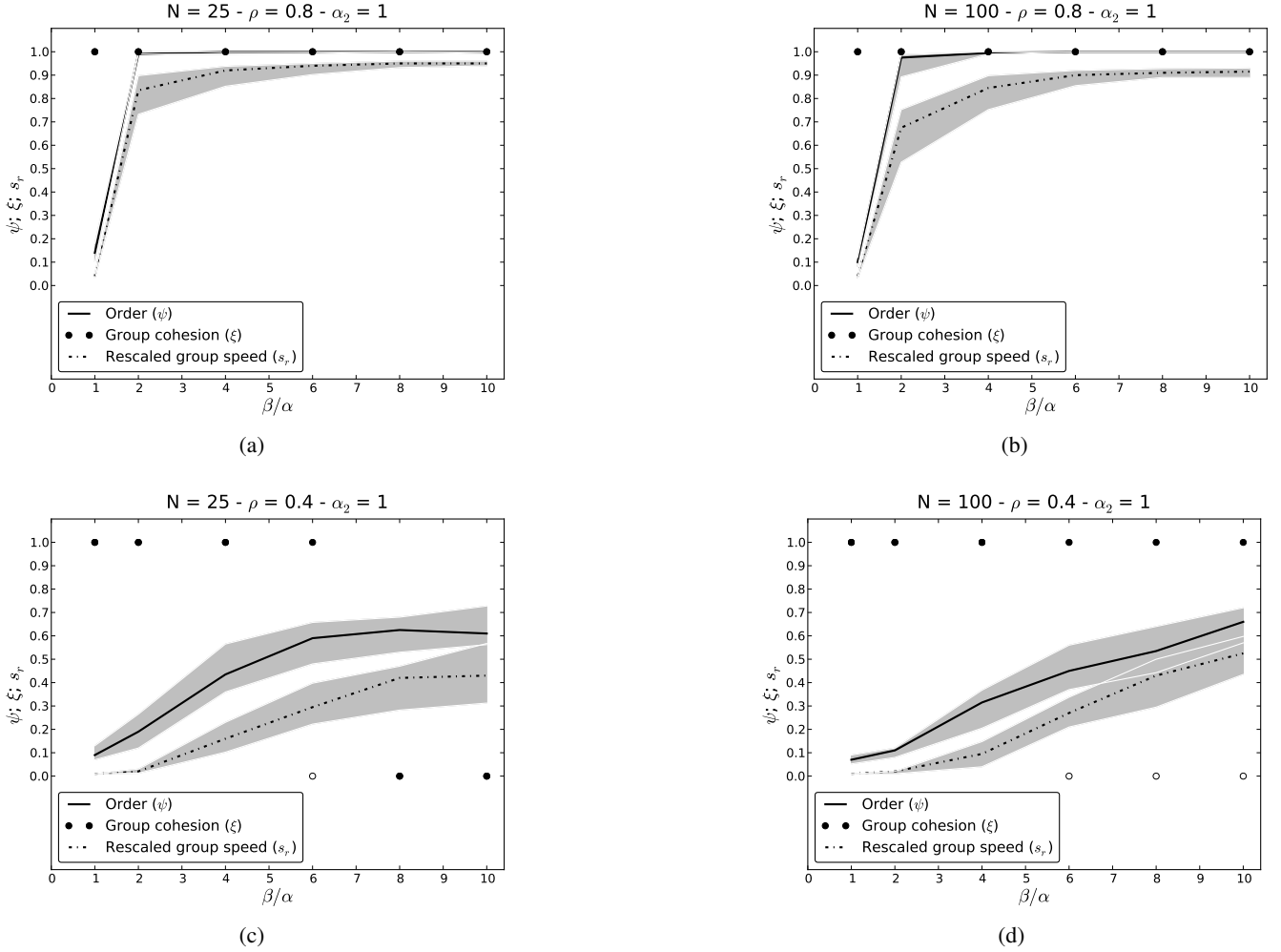


Figure 1: CMC-CMC case experiments for varying swarm size ( $N \in \{25, 100\}$ ) and ratio of aligning robots ( $\rho \in \{0.4, 0.8\}$ ). Thick lines show the median values, whereas the gray areas show the 25% and the 75% interquartile range of the data. For group cohesion, filled circles correspond to median values and empty circles to the 25% percentile score of the data.

## Results in the CMC-CMC case

The experimental results for CMC-CMC case are depicted in Figure 1. We first focus on the  $\rho = 0.8$  case, for both  $N = 25$  (Figure 1a) and  $N = 100$  (Figure 1b). Results show that the swarm is cohesive in most runs. However, order and speed are high only when  $\frac{\beta_1}{\alpha_1} \geq 2$ . Furthermore, while order is high at different values of the ratio  $\frac{\beta_1}{\alpha_1}$ , speed increases with increasing values of  $\frac{\beta_1}{\alpha_1}$ , until it saturates at around  $\frac{\beta_1}{\alpha_1} = 6$ . This shows that, when the alignment control vector is higher, robots tend to move faster. This is explained by the fact that the alignment control vector is more stable, over time, than the proximal control vector. Thus, the higher the weight of the alignment control vector, the more the robots tends to move forward rather than to turn. This allows the swarm to move faster, until speed saturates at the

maximum forward speed  $U$ .

When the proportion of aligning robots is  $\rho = 0.4$ , performance gets sensibly worse (Figures 1c and 1d). In both cases ( $N = 25$  and  $N = 100$ ), we observe two possible outcomes: for small values of the ratio  $\frac{\beta_1}{\alpha_1}$ , the swarm remains cohesive, but does not move. This happens because the relative contribution of the alignment control vector is not enough for the aligning robots to pull the entire swarm towards the agreed goal direction. For larger values of the ratio  $\frac{\beta_1}{\alpha_1}$ , group speed and order get higher. However, in at least 25% of the runs, the swarm splits. This happens because, in those runs, clusters of non-aligning robots are present. Since the motion of these robots is governed only by the proximal control vector, they are not able to match the higher speed of the aligning robots since they tend to turn more rather than to move forward, thus they remain disconnected from the group.

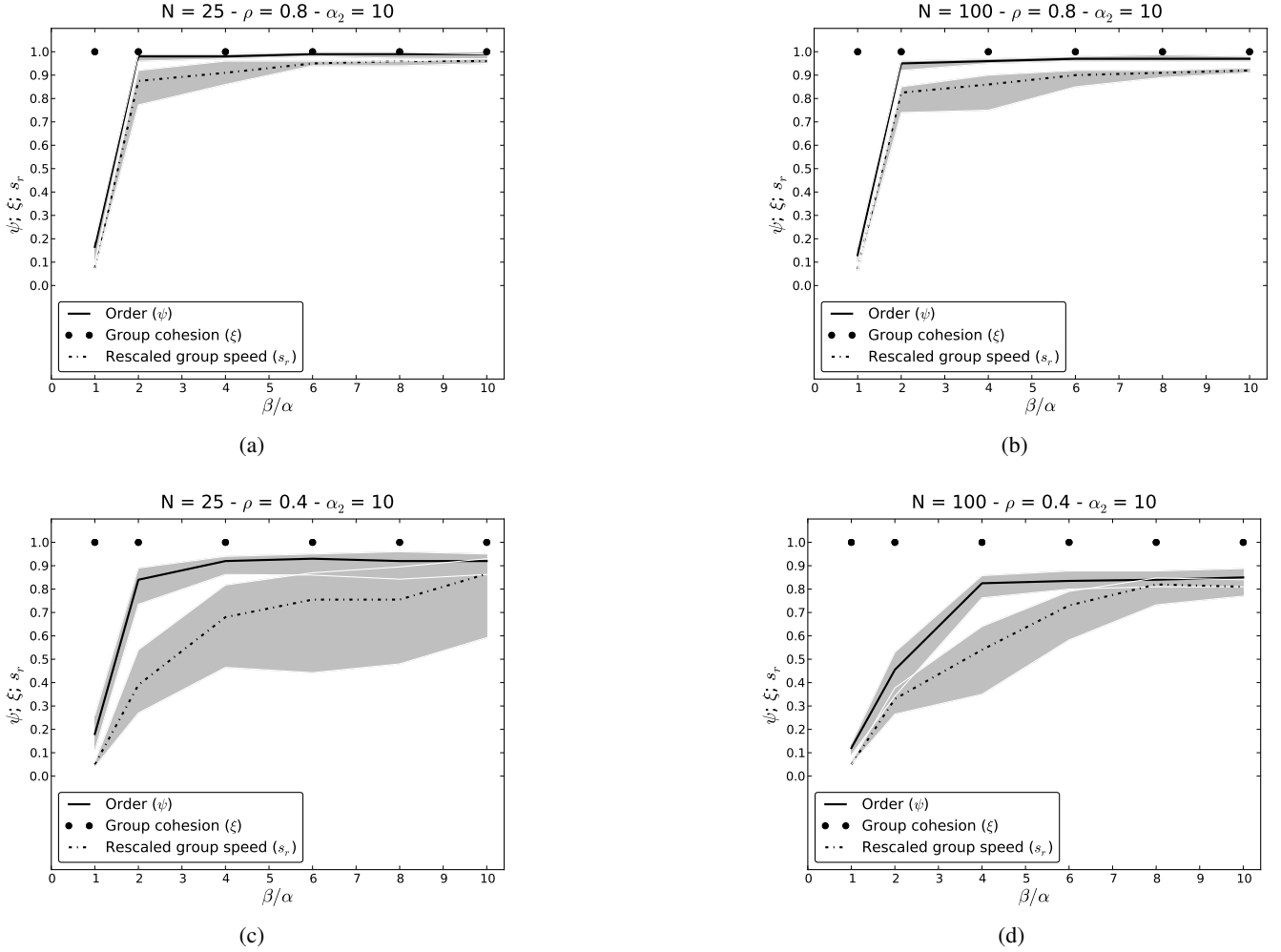


Figure 2: CMC-VMC case experiments for varying swarm size ( $N \in \{25, 100\}$ ) and ratio of aligning robots ( $\rho \in \{0.4, 0.8\}$ ). Thick lines show the median values, whereas the gray areas show the 25% and the 75% interquartile range of the data. For group cohesion, filled circles correspond to median values and empty circles to the 25% percentile score of the data.

In Stranieri et al. (2011), we also report the performance as function of  $\rho$ . We consider the case  $\frac{\beta_1}{\alpha_1} = 10$ , as it generally provides the best overall results. As shown in Stranieri et al. (2011), the flocking performance is acceptable in terms of the metrics used for  $\rho \geq 0.6$  in both cases  $N = 25$  and  $N = 100$ .

### Results in the CMC-VMC case

In the CMC-VMC case, results with  $\rho = 0.8$  (Figures 2a and 2b), are similar to the results obtained, with the same ratio, in the CMC-CMC case. The results with  $\rho = 0.4$  are much better in the CMC-VMC case (Figures 2c and 2d) with respect to the CMC-CMC case (Figures 1c and 1d). With both swarm sizes we have that, when  $\frac{\beta_1}{\alpha_1} > 2$ , the swarm is able to effectively flock together at the cost of a reduced speed.

In Stranieri et al. (2011), we also report the flocking performance as a function of  $\rho$  for  $\frac{\beta_1}{\alpha_1} = 10$  and  $\alpha_2 = 10$ . Differently from the CMC-CMC case, in the CMC-VMC case the performance of flocking degrades more gracefully as the proportion of non-aligning robots decreases.

The improved capability of the swarm to stay together is due to the advantage of using VMC in the non-aligning robots. In fact, non-aligning robots are able to respond to the high variations in the proximal control vector much more when they can also change their forward speed. As such, they are also able to stay together with the aligning robots, both when they are alone and when they are in small or big clusters. Finally, the reduced speed and the high variation of speed among runs is due to the following fact. In presence of a low proportion of aligning robots, we observed that the group heading direction is stable over short periods



of time but changes over long periods of time due to the disturbances caused by the non-aligning robots. This results in a non-linear trajectory executed by the entire swarm, which is different for each run. Since the rescaled group speed is computed assuming a linear trajectory, this measurement has large variation in the total displacement changes from run to run.

## CONCLUSIONS AND FUTURE WORKS

In this paper, we studied self-organized flocking in a swarm composed of behaviorally heterogeneous mobile robots. The swarm is composed of aligning robots, which are able to agree on a common heading direction, and non-aligning robots which lack this capability. We furthermore propose a new model for achieving motion in self-organized flocking. According to this model, aligning robots only change their angular speed, whereas non-aligning robots change both their forward and their angular speed.

We study the performance in terms of group alignment order, cohesiveness and speed. Results show that self-organized flocking is also possible when some individuals in the swarm lack the capability to agree on a common direction. More in particular, we showed that: i) a higher proportion of aligning robots always corresponds to a better performance; ii) performance is affected by the relative contribution of alignment and proximal control, and iii) for smaller proportions of aligning robots, flocking is possible only when the non-aligning robots also change their forward speeds.

Possible directions for future work are the following: First, we plan to study energy efficiency within the same framework of study. In particular, the use of a heterogeneous group of aligning and non-aligning robots poses a trade-off between efficiency of the motion and energy utilized. In fact, we observed that, in order for the swarm to hold cohesiveness, the non-aligning robots spend a lot of energy to vary their speed more reactively. Second, we would like to study the correlation between spatial aspects of the swarm composition. In particular, we would like to study whether particular configurations (i.e., topology, connectivity, ...) have different effects on the flocking performance. Third, we plan to perform experiments involving two different types of real robots.

## ACKNOWLEDGMENTS

This work was partially supported by the European Union through the ERC Advanced Grant “E-SWARM: Engineering Swarm Intelligence Systems” (contract 246939), and by the Meta-X project, funded by the Scientific Research Directorate of the French Community of Belgium. Alessandro Stranieri acknowledges support from the MIBISOC network, an Initial Training Network funded by the European Commission, grant PITN-GA-2009-238819. Carlo Pinciroli acknowledges support from ASCENS, a project

funded by the Future and Emerging Technologies programme of the European Commission. Mauro Birattari, and Marco Dorigo acknowledge support from the F.R.S.-FNRS of Belgium’s French Community, of which they are a Research Associate and a Research Director, respectively. The information provided is the sole responsibility of the authors and does not reflect the European Commission’s opinion. The European Commission is not responsible for any use that might be made of data appearing in this publication.

## References

- Ballerini, M., Cabibbo, N., Candelier, R., Cavagna, A., Cisbani, E., Giardina, I., V.Lecomte, Orlandi, A., Parisi, G., Procaccini, A., Viale, M., and Zdravkovic, V. (2007). Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the National Academy of Sciences of the United States of America*, 105(4):1232–1237.
- Bonani, M., Longchamp, V., Magnenat, S., Rétornaz, P., Burnier, D., Roulet, G., Vaussard, F., Bleuler, H., and Mondada, F. (2010). The MarXbot, a Miniature Mobile Robot Opening new Perspectives for the Collective-robotic Research. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, pages 4187–4193, Piscataway, NJ. IEEE Press.
- Buhl, J., Sumpter, D. J. T., Couzin, I., Hale, J., Despland, E., Miller, E., and Simpson, S. J. (2006). From disorder to order in marching locusts. *Science*, (312):1402–1406.
- Camazine, S., Franks, N., Sneyd, J., Bonabeau, E., Deneubourg, J.-L., and Theraulaz, G. (2001). *Self-Organization in Biological Systems*. Princeton University Press, Princeton, NJ.
- Çelikkanat, H. and Şahin, E. (2010). Steering self-organized robot flocks through externally guided individuals. *Neural Computing and Applications*, 19(6):849–865.
- Couzin, I. D., Krause, J., Franks, N. R., and Levin, S. A. (2005). Effective leadership and decision-making in animal groups on the move. *Nature*, 433:513–516.
- Couzin, I. D., Krause, J., James, R., Ruxton, G. D., and Franks, N. R. (2002). Collective memory and spatial sorting in animal groups. *Journal of theoretical biology*, 218:1–11.
- Diwold, K., Schaerf, T. M., Myerscough, M. R., Middendorf, M., and Beekman, M. (2011). Deciding on the wing: in-flight decision making and search space sampling in the red dwarf honeybee *Apis florea*. *Swarm Intelligence*, 5(2):121–141.
- Dyer, J. R. G., Ioannou, C. C., Morrell, L. J., Croft, D. P., Couzin, I. D., A., W. D., and Krause, J. (2008). Consensus decision making in human crowds. *Animal Behavior*, 75(2):461–470.
- Ferrante, E., Turgut, A. E., Mathews, N., Birattari, M., and Dorigo, M. (2010). Flocking in stationary and non-stationary environments: A novel communication strategy for heading alignment. In Schaefer, R., Cotta, C., Kołodziej, J., and Rudolph, G., editors, *Parallel Problem Solving from Nature – PPSN XI*, volume 6239 of *Lecture Notes in Computer Science*, pages 331–340. Springer.

- Graves, G. R. and Gotelli, N. J. (1993). Assembly of avian mixed-species flocks in amazonia. *Proceedings of the National Academy of Sciences of the United States of America*, 90(4):1388–1391.
- Hayes, A. and Dormiani-Tabatabaei, P. (2002). Self-organized flocking with agent failure: Off-line optimization and demonstration with real robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 3900–3905, Piscataway, NJ. IEEE Press.
- Hettiarachchi, S. and Spears, W. (2009). Distributed adaptive swarm for obstacle avoidance. *International Journal of Intelligent Computing and Cybernetics*, 2(4):644–671.
- Holland, O., Woods, J., Nardi, R., and Clark, A. (2005). Beyond swarm intelligence: the ultraswarm. In *Proc. of the IEEE Swarm Intelligence Symposium*, pages 217–224, Piscataway, NJ. IEEE Press.
- Janson, S., Middendorf, M., and Beekman, M. (2005). Honeybee swarms: how do scouts guide a swarm of uninformed bees? *Animal Behaviour*, 70(2):349–358.
- Kelly, I. and Keating, D. (1996). Flocking by the fusion of sonar and active infrared sensors on physical autonomous robots. In *Proc. of the Third Int. Conf. on Mechatronics and Machine Vision in Practice*, pages 14–17.
- Krause, J. (1993). The relationship between foraging and shoal position in a mixed shoal of roach (*rutilus rutilus*) and chub (*leuciscus cephalus*): a field study. *Oecologia*, 93:356–359.
- Krause, J., Reeves, P., and Hoare, D. (1998). Positioning behaviour in roach shoals : The role of body length and nutritional state. *Behavior*, 135(8-9):1031–1039.
- Matarić, M. J. (1994). *Interaction and Intelligent Behavior*. PhD thesis, MIT, MA.
- Michelena, P., Jeanson, R., Deneubourg, J.-L., and Sibbald, A. M. (2010). Personality and collective decision-making in foraging herbivores. *Proceedings of the Royal Society B*, 277:1093–1099.
- Momen, S., Amavasai, B. P., and Siddique, N. H. (2007). Mixed species flocking for heterogeneous robotic swarms. In *The International Conference on Computer as a Tool (EUROCON 2007)*, pages 2329 – 2336, Piscataway, NJ. IEEE Press.
- Moslinger, C., Schmickl, T., and Crailsheim, K. (2009). A minimalist flocking algorithm for swarm robots. In *European Conference of Artificial Life (ECAL 2009)*.
- Pinciroli, C., Trianni, V., O’Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Caro, G. D., Ducatelle, F., Stirling, T., Gutiérrez, A., Gambardella, L. M., and Dorigo, M. (2011). ARGoS: a modular, multi-engine simulator for heterogeneous swarm robotics. Technical Report TR/IRIDIA/2011-009, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium.
- Reebs, S. G. (2000). Can a minority of informed leaders determine the foraging movements of a fish shoal. *Animal Behavior*, 59:403–409.
- Reynolds, C. (1987). Flocks, herds and schools: A distributed behavioral model. In Stone, M. C., editor, *SIGGRAPH ’87: Proc. of the 14th annual conference on computer graphics and interactive techniques*, pages 25–34, New York. ACM Press.
- Roberts, J., Stirling, T., Zufferey, J., and Floreano, D. (2009). 2.5d infrared range and bearing system for collective robotics. In Papanikolopoulos, N., Sugano, S., Chiaverini, S., and Meng, M., editors, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Piscataway, NJ. IEEE Press.
- Sayama, H. (2009). Swarm chemistry. *Artificial Life*, 15(1):105–114.
- Spears, W. M., Spears, D. F., Hamann, J. C., and Heil, R. (2004). Distributed, physics-based control of swarms of vehicles. *Autonomous Robots*, 17:137–162.
- Stranieri, A., Ferrante, E., Turgut, A. E., Trianni, V., Pinciroli, C., Birattari, M., and Dorigo, M. (2011). Oself-organized flocking with a heterogeneous mobile robot swarm. Supplementary information page at <http://iridia.ulb.ac.be/supp/IridiaSupp-2011-011/>.
- Turgut, A. E., Çelikkanat, H., Gökçe, F., and Şahin, E. (2008). Self-organized flocking in mobile robot swarms. *Swarm Intelligence*, 2(2):97–120.
- Vicsek, T., Czirok, A., Ben-Jacob, E., Cohen, I., and Shochet, O. (1995). Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75(6):1226–1229.

# An improved transfer entropy method for establishing causal effects in synchronizing oscillators

James Thorniley

Center for Computational Neuroscience and Robotics  
University of Sussex, Brighton, UK  
jt241@sussex.ac.uk

## Abstract

A number of authors have proposed that the analysis of spatio-temporal information transfer may help to understand cognitive behavior in biological and artificial agents. A specific case of interest is the study of synchronization of central pattern generators (CPGs) in embodied systems. Pitti et al. used simulated biped walkers to demonstrate a correlation between task success and measured transfer entropy from the body to the neural oscillator. This suggests it may be possible to use transfer entropy to help understand, control and improve the behavior of limbed robots.

This paper presents a novel method of analyzing synchronizing oscillators with transfer entropy, which it is hoped will lead to advances in controlling such systems. The necessary discretization of continuous time oscillator observations is performed via a stroboscopic analysis that preserves the information of interest and allows a natural interpretation of the results. Unlike some CPG studies, the current work addresses the tendency of naive transfer entropy calculations to overestimate causal relationships in a significant and non-trivial way. Transfer entropy may also underestimate causal links when used on purely observational data, so it is important to determine the limits of the method. It is found that in weak (rather than rigid) synchronization transfer entropy can be measured and interpreted as a causal information flow.

## Introduction

Recent work has developed information theoretic understandings of embodied cognition (Lungarella and Sporns, 2006; Pfeifer et al., 2007). In learning and adaptation, information theoretic feedback mechanisms can guide system development in the absence of explicit goal directed feedback (Klyubin et al., 2005). By studying the information flow in an evolved artificial agent's neural network, Williams and Beer (2010) are able to explain how the neural dynamics perform a computational role. In evolutionary robotics, the use of information theoretic goal functions has been shown to generate interesting behaviors (Der et al., 2008).

Central pattern generator (CPG) synchronization is thought to underlie animal gaits (Collins and Stewart, 1993) and potentially therefore has applications in robotics. However, in a complex synchronizing system it is often difficult

to determine the nature of the interactions between oscillators. Ceguerra et al. (2011) approached this problem by using a form of transfer entropy to analyze the synchronization process in networks of coupled oscillators, which was shown to be more effective than other methods.

The current work investigates information transfer between oscillators coupled by non-trivial physical mechanisms. This is similar to the study by Pitti et al. (2009), in which simulated biped walkers were coupled to oscillators. It was shown that at optimal values of the coupling (where the best walking behavior is achieved) there is an increase in information transfer from the body to the oscillator. Thus the information transfer is thought to correlate to the successful entrainment of the body and controller dynamics.

Transfer entropy (Schreiber, 2000) is the information gained from conditioning the entropy rate of a time dependent variable on a secondary historical variable as well as its own past. It is a directional measure, and is often interpreted as signifying causal links (Pitti et al., 2009; Lungarella and Sporns, 2006), however when used to analyze finite experimental time series data there is a strong risk of overestimating such causal influence, a problem that is known from the literature (Marschinski and Kantz, 2002; Lizier and Prokopenko, 2010). Furthermore, because transfer entropy is applied here to purely observational data, this method is only claimed to be a reasonable guide for detecting causal influences, as outside intervention would be needed to expose all causal links (Ay and Polani, 2008). A complete scientific approach to building causal models is well beyond the scope of this paper (see Pearl, 2009).

The method of calculating transfer entropy presented here uses a novel time discretization approach, inspired by the stroboscopic analysis of Schäfer et al. (1998). Other than that, the method follows Marschinski and Kantz (2002) by conditioning on the longest practicable history of the target variable (a requirement that is sometimes neglected).

The effect of oscillator coupling on transfer entropy is investigated in a continuous time system composed of either a single oscillator and passive body model, or a pair of two such systems (see Figure 1). The following sections will first

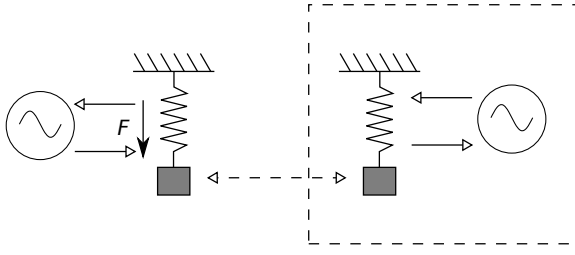


Figure 1: Simplified system under study. An electronic oscillator is attached to a mass-spring-damper system, providing force actuation and incorporating the resultant extension of the spring into the feedback path of the oscillator circuit. The model is extended by duplicating the system (dashed box) and coupling via the mechanical component.

introduce the models studied, and later develop the transfer entropy calculation based on time series data from simulations of these models. The relationship between transfer entropy and the dynamical process of synchronization is discussed. It is argued that a state of synchronization will not always lead to increased transfer entropy, but during an ongoing process of weak synchronization transfer entropy will be found, and in such cases will show causal relationships. It appears that measured transfer entropy may be useful in predicting the effects of making changes in a system, such as varying coupling parameters, which implies that it may be possible to develop control techniques based on information transfer.

## Model construction

The model developed here is intended to be a minimal first approximation to a physically realizable modular active dynamic walker. That is, though it is intended that further work will develop this system as a real robot, the current model is not fully realistic, but contains the fundamental components: a chaotic oscillator that can be implemented as an electronic circuit, and a simple mass-spring-damper system analogous to a passive compliant robot body. This structure makes it comparable to the architecture of Pitti et al. (2009), except that the neural controller here is a continuous time analog circuit, rather than a discrete time map, and the physical component does not include a full environment.

### Chaotic oscillator

The oscillator design developed by Sprott (2000) and improved on in Kiers et al. (2004) was chosen. It is extremely simple to implement using widely available electronic components, and can easily be tuned to produce chaotic or periodic behavior. The Sprott system has the further advantage of having relatively stable dynamics, in that it tends to reach its only periodic or chaotic attractor quickly after being biased at a sensible voltage and does not tend to drift or diverge away from the desired dynamics.

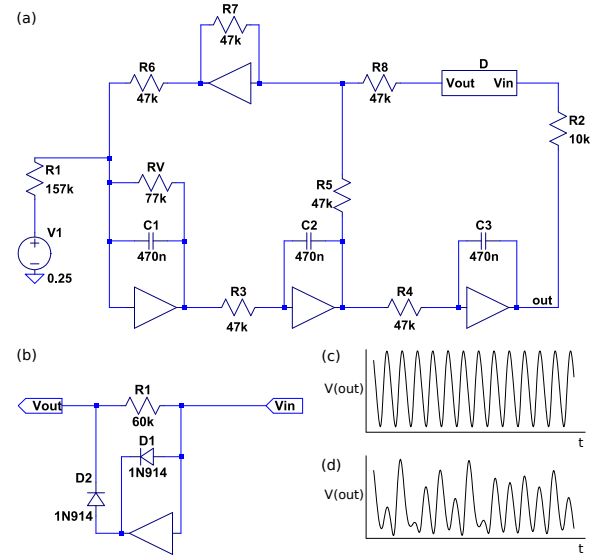


Figure 2: Chaotic oscillator based on Kiers et al. (2004). (a) Positive feedback based circuit design. (b) Nonlinear sub-circuit component  $D$ . (c) Typical steady state periodic output simulated using LTSpice, with resistor  $R_v$  set to 50 kΩ (periodic solution) and (d) 77 kΩ (chaotic solution).

The dynamics of the oscillator in isolation are well documented by Kiers et al. (2004), but it is useful to recap them here. The circuit diagram is given in Figure 2. The circuit consists of three op-amp integrators producing anti-derivative signals in a chain, with the output of the final integrator being fed back through a nonlinear amplifier and a combined with the output of the other integrators with a summing amplifier. The circuit effectively implements the following third-order “jerk” function in which  $D(x) = -6 \min(x, 0)$  and  $Q$  and  $\alpha$  are constants.

$$\ddot{x} = -Q\ddot{x} - \dot{x} + D(x) - \alpha \quad (1)$$

The constant bias  $\alpha$  is provided by the voltage source, and for a range of small positive values (e.g.  $\alpha = 0.1$  works well) it will allow oscillatory behavior. Tuning  $R_v$  in the circuit will vary the  $Q$  parameter, which will result in chaotic and periodic solutions at different values, as shown in Figures 2c and d, generated by simulating the circuit with LTSpice<sup>1</sup>. The fundamental frequency of the oscillations is related to the time constants of the integrators, that is  $\frac{1}{\omega_0} = \tau_0 = RC = 47 \text{ k}\Omega \times 470 \text{ nF} \approx 0.022 \text{ s}$  in the example circuit of Figure 2. Equation 1 is a non-dimensionalized description of the circuit attained by taking the derivatives with respect to  $\omega_0 t$  (so if  $t$  is in seconds, equation 1 will effectively have a natural frequency of  $\omega_0 = 1 \text{ rad s}^{-1}$ ). Thus control of the fundamental frequency in the numerical solution of equation 1 is achieved, when required, by rescaling

<sup>1</sup><http://www.linear.com/designtools/software/#LTspice>

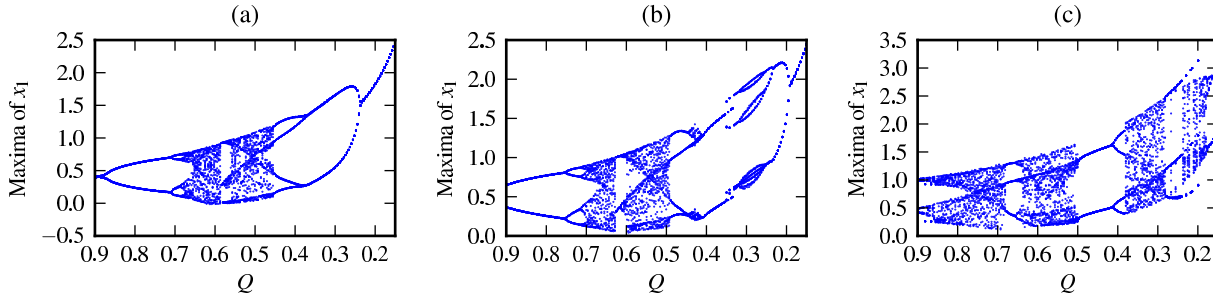


Figure 3: Maxima of the oscillator variable  $x_1$  at different values of  $Q$ . The fixed parameters are  $\alpha = 0.1$ ,  $\zeta = 0.3$  and the coupling is (a)  $\gamma = 0$  (no coupling), (b)  $\gamma = 0.1$ , (c)  $\gamma = 0.3$

the time variable by the desired value of  $\omega_0$ .

Figure 3a shows the effect of  $Q$  on the dynamics of the system, by showing the maxima of  $x_1$  over time at different values of  $Q$ . The diagram can be obtained by numerically integrating equation 1 with a computer library such as LSODE (Hindmarsh, 1983), as was used here, or by using a SPICE simulation of the circuit in Figure 2. The system follows a period doubling route to chaos as  $Q$  decreases, with a notable periodic island around  $Q \approx 0.58$  and returns to periodicity via a further bifurcation near  $Q \approx 0.47$ .

### Coupled mass-spring-damper

The dynamics of an ideal mass-spring-damper (MSD for brevity) can be expressed in terms of the time dependent extension of the spring  $x$  using the second order differential equation 2 with  $m$  being the mass,  $k$  the spring constant and  $c$  the damping coefficient.

$$\ddot{x} + \frac{c}{m}\dot{x} + \frac{k}{m}x = 0 \quad (2)$$

Alternatively define the angular velocity  $\omega_0 = \sqrt{\frac{k}{m}}$  and the damping ratio  $\zeta = \frac{c}{2\sqrt{mk}}$ , take derivatives with respect to  $\omega_0 t$  as in equation 1 (see above) and rearrange to get:

$$\ddot{x} + 2\zeta\dot{x} + x = 0 \quad (3)$$

To couple the two systems together, the oscillator variable is added to the acceleration of the spring system after subtracting 0.5 (to make the influence of the oscillator approximately symmetric around zero, as it normally oscillates between around 0 and 1), and the spring extension is added to the feedback path of the oscillator after multiplying by a coupling parameter  $\gamma$ . Thus the complete system is given by equations 4 and 5, with  $x_1$  and  $x_2$  being the time varying oscillator variable and spring extension respectively.

$$\ddot{x}_1 = -Q\ddot{x}_1 - \dot{x}_1 + D(x_1 + \gamma x_2) - \alpha \quad (4)$$

$$\ddot{x}_2 = -2\zeta\dot{x}_2 - x_2 + (x_1 - 0.5) \quad (5)$$

Note that the derivatives in both equations are taken with respect to the same time variable  $\omega_0 t$  and thus the oscillator and MSD systems always have identical natural angular velocities. This implies  $RC = \sqrt{\frac{m}{k}}$ , clearly in a real system this would be unlikely, so the following assumes that small discrepancies in the spring and oscillator natural frequencies are of little consequence.

The coupling could be achieved electronically by adding a voltage signal into the input of the nonlinear feedback amplifier in the circuit in Figure 2 via a series variable resistor such that when the diodes are switched off the op-amp in the nonlinear sub-circuit  $D$  acts as a summing amplifier and the variable resistor allows control of the coupling strength.

With no coupling ( $\gamma = 0$ ) clearly the oscillator drives the MSD but will not be influenced by it, thus the bifurcations of the oscillator dynamic will remain as in Figure 3a. With  $\gamma > 0$  the bifurcation structure changes dramatically, as shown in Figures 3b and 3c.

### Synchronization vs. resonance

Is the change of the dynamics as coupling is increased shown in Figure 3 a form of synchronization? When coupling (here meaning the feedback coefficient  $\gamma$ ) is zero an engineer might call the MSD system a passive resonant filter – remember that is it still driven by the electronic oscillator, so the frequency spectrum of the spring extension will appear to be a filtered version of the oscillator output. With feedback however, the oscillator changes its behavior noticeably as we have seen, so perhaps there is something more than simple resonance happening – a form of synchronization.

This appears to be the view taken by Pitti et al. (see Figure 3 in Pitti et al., 2009), who suggest that resonance is the forward process (from oscillator to dissipative system) and synchronization occurs along the feedback path. However this seems to contradict the view of Pikovsky et al. (2001, pp. 14–17) and Ceguerra et al. (2011), who require that synchronization only applies to synchronous variation of systems that are capable of oscillating independently.

Pikovsky et al. give the example of an ecological sys-

tem such as hare-lynx populations, where both variables (the populations of the two species) oscillate in a phase locked manner, but the system cannot be decomposed into isolated subsystems. Assuming the lynxes eat only hares, then an isolated lynx population with no access to hares would simply die out, not oscillate at some natural frequency. Likewise an isolated mass-spring-damper not stimulated by an appropriate oscillator will die down as its energy dissipates.

Of course some mechanical systems can oscillate independently – think of a passive dynamic walker (McGeer, 1990), a biped structure that walks down a hill, obtaining its energy purely from gravitational potential. As long as the slope is present, the passive dynamic walker could be considered to have its own natural oscillation. If this were the system being coupled to a neural oscillator, then it would seem that true synchronization could be discussed. However current scenario of an isolated mass-spring-damper is unambiguous – there is decomposition that leaves two oscillators and hence no synchronization as a “complex dynamical process, not a state” (Pikovsky et al., 2001). In the later experiments of Pitti et al. (2009) there are multiple neural oscillators in a single system, so the notion of synchronization becomes more applicable. This scenario will also be investigated later in this paper.

## Transfer entropy

This section will develop a measure of transfer entropy that can be meaningfully applied to continuous time oscillators.

Initially, assume we have two discrete time series  $X$  and  $Y$  of finite length. The value of  $X$  at time  $n \in \{1, 2, \dots, N\}$  is  $X_n$ , discretized such that  $X_n \in \mathbf{X}$ , a finite set of symbols. Apply the same notation to  $Y$ .

The  $k$ -history of  $X$  at  $n$ , i.e.  $\{X_n, X_{n-1}, \dots, X_{n-k+1}\}$  is written  $X_n^{(k)}$ , and likewise the  $l$ -history of  $Y$  is  $Y_n^{(l)}$ . The transfer entropy from series  $Y$  to series  $X$ , written  $T_{Y \rightarrow X}$ , is the information gained about  $X_{n+1}$  in moving from prior knowledge of  $X_n^{(k)}$  alone to also having  $Y_n^{(l)}$ . This is given by the Kullback-Leibler divergence, or equivalently the conditional mutual information, calculated using the summation in equation 6.

$$\begin{aligned} T_{Y \rightarrow X} &= I(X_{n+1}; Y_n^{(l)} | X_n^{(k)}) \\ &= D_{KL}(P(X_{n+1} | X_n^{(k)}, Y_n^{(l)}) || P(X_{n+1} | X_n^{(k)})) \\ &= \sum_{x'} P(X_{n+1}, X_n^{(k)}, Y_n^{(l)}) \log \frac{P(X_{n+1} | X_n^{(k)}, Y_n^{(l)})}{P(X_{n+1} | X_n^{(k)})} \end{aligned} \quad (6)$$

The probabilities are estimated from observations of a single instance over a long time series. This is similar to the methods of Pitti et al. (2009) and Marschinski and Kantz (2002). It is very important therefore that the time series is statistically stationary over the period of interest, which can be a practical problem with transfer entropy calculations.

It is also possible to calculate similar information transfer statistics from ensembles of non-stationary systems by calculating probabilities from the ensemble at each point in time (e.g. Ceguerra et al., 2011; Williams and Beer, 2010). The time average approach was chosen here because there is at least potential applicability to complex real systems (e.g. a real robot) where experiments cannot be repeated in such a way that the entropy calculation would be possible and valid.

The following sections consider further practical issues regarding the application of this measure to the simulated time series in these experiments. Since these time series are continuous, sensible discretizations must be established. Further, appropriate values of  $k$  and  $l$  need to be chosen.

## Stroboscopic discretization

The time series are first analyzed using a stroboscopic method similar to that of Schäfer et al. (1998). Consider again the continuous time series generated by the coupled oscillator-MSD system in equations 4 and 5:  $x_1(t)$  and  $x_2(t)$ . The series of maxima of the oscillator voltage are  $\hat{x}_1$  and the time of the  $n$ th maximum of  $x_1$  is  $\hat{t}(n; x_1)$ . Figure 4 shows the phase of  $x_2$  plotted at each maximum of  $x_1$ .

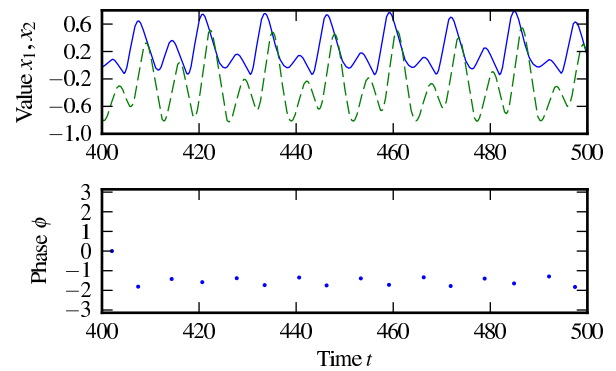


Figure 4: Stroboscopic visualization of the spring extension at each maximum of the oscillator in the coupled oscillator-MSD system. Top: The solid blue line is the oscillator voltage  $x_1$ , and the dashed green line is the spring extension  $x_2$ . Bottom: The points represent the phase of  $x_2$  taken at each maximum of  $x_1$ . The oscillator is set to a chaotic mode with  $Q = 0.67$ , the remaining system parameters are  $\alpha = 0.1$ ,  $\omega_0 = 1 \text{ rad s}^{-1}$ ,  $\zeta = 0.3$  and  $\gamma = 0.01$ .

The phase of the spring extension is calculated here on a “peak-to-peak” basis. That is, the phase of the spring at the  $n$ th maximum of the oscillator  $x_1$  is taken to be the (linear) proportion of the time between the last and the next maximum of the spring extension  $x_2$  that has already elapsed, written  $\phi(n; x_2, x_1)$ . Alternative methods of calculating the phase were considered, such as using the Hilbert transform as per Schäfer et al. (1998). This was found to be problematic due to the chaotic nature of the signals here, hence the

simpler peak-to-peak method was chosen, but in other applications the Hilbert transform might well provide useful phase values to use with this transfer entropy method.

For each phase angle of  $x_2$ ,  $\phi(n; x_2, x_1)$  with  $n > 1$ , the time period of the most recent oscillation of  $x_1$  can also be calculated by:

$$\Delta\hat{t}(n; x_1) = \hat{t}(n; x_1) - \hat{t}(n-1; x_1) \quad (7)$$

Thus the two continuous time series  $x_1$  and  $x_2$  can be converted to “stroboscopic” discrete time representation:  $\Delta\hat{t}(n; x_1)$  and  $\phi(n; x_2, x_1)$ , usually defined for all  $n \in \{2, 3, \dots, N\}$  (with the proviso that  $\phi$  is only defined when the nearest maxima of  $x_2$  are known).

It would of course have been possible to simply discretize  $x_1$  and  $x_2$  by choosing arbitrary time intervals. The advantage of the stroboscopic method is that the time intervals are determined naturally by the dynamics of the system. Furthermore, the time series being compared here have a natural interpretation in terms of synchronization, which is well documented in the literature. In what follows the “stroboscopic” transfer entropy is effectively the influence of the phase difference on the future frequency of oscillation. This will be denoted  $ST_{A \rightarrow B}$  as shorthand for transfer entropy after the stroboscopic conversion has been applied, i.e.  $T_{\phi(\cdot; A, B) \rightarrow \Delta\hat{t}(\cdot; B)}$ . so this measure is similar to that of Ceguerra et al. (2011), except that here the timing of the samples is based on the system’s oscillation rather than arbitrary. In other words, the entropy is measured in bits per oscillation rather than bits per second.

## Simulation method

The “stroboscopic” time series defined above can easily be obtained from numerical simulation of the oscillator-MSD system. First LSODE was used to obtain a solution to the initial value problem given by equations 4 and 5 via numerical integration, with the starting values of  $x_1$ ,  $x_2$  and the necessary derivatives ( $\dot{x}_1$  etc) at time  $t_0 = 0$  chosen randomly from the range  $[0, 1)$ . The first part of the time series from  $t_0$  to a chosen cutoff point  $t_{tr}$  was discarded to remove the “transient” dynamic of the system. During the following interval, between  $t_{tr}$  and the end of the simulation at another chose time  $t_1$ , it is assumed that the dynamic reaches an attractor (observation suggests that this is the case). Therefore the observations used should be at least nearly statistically stationary.

Inevitably the numerical solution of the equations will give a discrete-time output series, with intervals chosen here to be one twentieth of a (simulated) second between points. Thus oscillations have a period of around 120 simulation intervals (recall that the effective time constant of the system was  $2\pi$ ). The maximum times ( $\hat{t}$ ) were estimated by finding those values in the simulated time series that were preceded and followed by lower values – a crude method but it is effective in this case.

To estimate the necessary probability distributions, the frequencies of samples in  $p$  bins was used, with the bin sizes adjusted such that each has a similar number of data points in it, following Marschinski and Kantz (2002). More advanced methods could be applied but for the current purposes this appeared to be adequate and should produce reliable results.

## Causality and transfer entropy

Transfer entropy can sometimes be thought of as “causal information”, but care needs to be taken. Transfer entropy is literally, from equation 6, an information gain in moving from conditioning the future of  $X$  on its own history  $X^{(k)}$  alone to conditioning on the joint history of  $X^{(k)}$  and  $Y^{(l)}$ . Suppose that  $k$  and  $l$  (the history lengths) are both 1, as is sometimes the case in the literature (Pitti et al., 2009; Lungarella and Sporns, 2006). The immediate history  $Y^{(1)}$  can contain information about the future states of  $X$  without having any real causal influence if it contains information about past states of  $X$  that have not been conditioned for in the mutual information calculation – i.e.  $X^{(k)}$  is too short a past.

This problem is clearly a possibility in the system under study here. In a single oscillator-MSD system with the feedback coupling  $\gamma$  set to 0, we know from the design of the system that the spring extension has no influence on the oscillator dynamic, but we also know that the MSD system stores mechanical energy and hence contains information about its own past and the past of the oscillator (which stimulated it). Thus the current state of the spring may help to predict the future state of the oscillator when added to just the current state or recent past of the oscillator, but if we control for the entire history of the oscillator, the spring state cannot be useful, as it is itself determined entirely by the history of the oscillator.

Marschinski and Kantz (2002) present a method designed to minimize this overestimation: set  $l$  to 1, then increase  $k$  from 1 until any causal influence of  $X_n^{(k)}$  on the future  $X_{n+1}$  is already accounted for before calculating the information gained by including  $Y_n^{(1)}$ . However, increasing  $k$  will rapidly expand  $\mathcal{X}$  in equation 6, i.e. the support set for which probabilities must be found. In a finite time series this will result in fewer examples of each combination of states from which to calculate the conditional probabilities, and ultimately to sometimes significant overestimation of the transfer entropy. The solution proposed by Marschinski and Kantz is to subtract the transfer entropy obtained when the  $Y$  series is randomly permuted in time, so that any true temporal correlations are lost and any calculated transfer entropy must therefore be due to finite sample error. This measure (which they called effective transfer entropy) will be used in all calculations to follow.

Figure 5a shows how effective transfer entropy overestimates the causal influence of the spring on the oscillator when  $k = 1$  even when no feedback coupling is present so



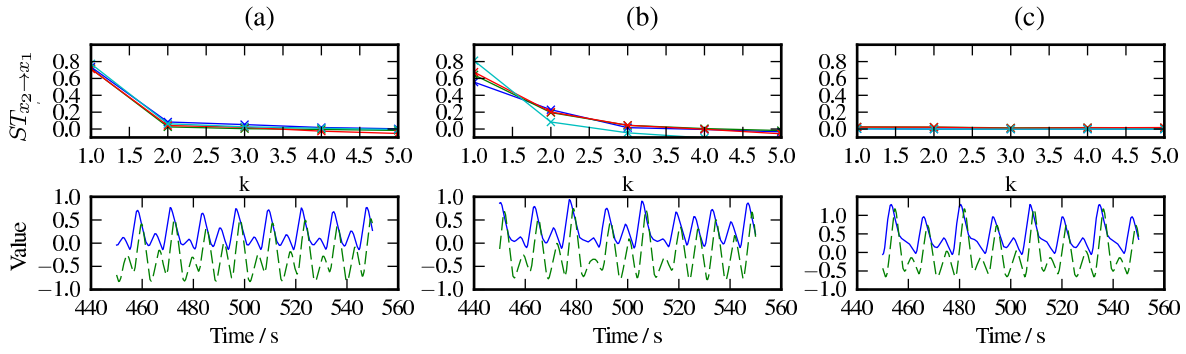


Figure 5: Stroboscopic effective transfer entropy for a single coupled oscillator-MSD system. Coupling was (a)  $\gamma = 0$  (no coupling) (b)  $\gamma = 0.1$  (c)  $\gamma = 0.3$ . Top row:  $ST_{x_2 \rightarrow x_1}$  (from spring phase to the oscillator periods), using four partition sizes  $p \in \{4, 5, 6, 7\}$ . Bottom row: short sections of simulated time series. Other parameters:  $\omega_0 = 1 \text{ rad s}^{-1}$ ,  $\zeta = 0.3$ ,  $Q = 0.67$ ,  $\alpha = 0.1$ . Time series analysed between  $t_{tr} = 400 \text{ s}$  and  $t_1 = 15000 \text{ s}$  with measurement interval  $\Delta t = \frac{1}{20} \text{ s}$ .

the spring cannot possibly have causal influence on the oscillator. Furthermore, it appears that the overestimate at small  $k$  is the only source of apparent transfer entropy even when coupling is added: Figure 5b,  $\gamma = 0.1$ ,  $ST_{x_2 \rightarrow x_1}$  rapidly declines when  $k \geq 2$ . Though the spring does influence the dynamics of the coupled system as we know, this cannot be detected from the time series as the spring can only store information previously generated by the oscillator. When coupling is higher ( $\gamma = 0.3$ , Figure 5c) the system reaches a limit cycle dynamic, and no transfer entropy is detected at any point. The fact that the spring is not an independent oscillator implies both that this is not a true process of synchronization, and further than no transfer entropy can be measured.

### Transfer between two oscillators

The above results show that for transfer entropy to be present with larger values of  $k$  (i.e. to genuinely signify causal dependence), there must be at least two systems capable of producing information. This can be achieved by duplicating the oscillator-MSD system and coupling via the mechanical component (the MSD). There are therefore two  $x_1$  and  $x_2$  variables (one for each system), the coupling is added by updating equation 5 (the dynamics of the MSD) to add the difference between the two springs multiplied by a coupling coefficient  $\gamma_c$  to the acceleration of the local spring. Equation 8 therefore gives the acceleration of the local spring  $\ddot{x}_2$  given the remote spring extension  $x'_2$ .

$$\ddot{x}_2 = -2\zeta\dot{x}_2 - x_2 + (x_1 - 0.5) + \gamma_c(x'_2 - x_2) \quad (8)$$

This system is now a loose analog of a pair of mechanically coupled neural-mechanical systems, call them system  $A$  and system  $B$ . In what follows, system variables and parameters will be superscripted with an  $A$  or  $B$  to signify (where it is ambiguous) which system they are a part of,

e.g.  $x_1^A$  and  $x_1^B$  are the oscillator values for system  $A$  and system  $B$  respectively. Usually the parameters will be identical in both systems, in which case a superscript is not used. The introduced coupling  $\gamma_c$  models the mechanical linkage between the two systems, whereas the existing internal coupling  $\gamma$  can be viewed as a kind of proprioceptive feedback to an oscillator from the limb it directly controls. The following experiments will study the effect of changing  $\gamma$  while keeping  $\gamma_c$  constant, i.e. to ask the question: *given a fixed mechanical coupling (body morphology), how do changes in internal coupling affect overall synchronization, information transfer and causal influences.*

Figure 6 shows how increasing values of internal coupling  $\gamma$  (with the spring coupling fixed at  $\gamma_c = 10$ ) affect the ratio of the mean peak-to-peak frequencies of oscillator  $B$  and oscillator  $A$ , i.e.  $\frac{f_B}{f_A}$  where  $f^S = \langle \Delta t(\cdot; x_1^S) \rangle^{-1}$ . The natural frequency of system  $A$  is fixed at  $\omega_0^A = 1 \text{ rad s}^{-1}$  and  $\omega_0^B$  is varied as shown. Clearly for no coupling the frequencies should vary independently, so the observed frequency difference varies linearly with the natural frequency difference. As coupling is increased, the observed frequencies appear to be pushed further from the natural frequency difference, until at coupling greater than  $\gamma \approx 0.3$ , a synchronization region appears around the central part of the plot where the frequencies tend to lock to a 1:1 ratio (except just around  $\gamma = 0.45$  where there are two peaks representing areas where the frequency ratio, though still synchronized, is 3:2). As  $\gamma$  approaches 0.5 this region starts to shrink again, suggesting an optimal value of  $\gamma$  (in terms of the likelihood of frequency locking) exists in this region.

The effective transfer entropy when the coupling between the oscillators is  $\gamma = 0.35$  is shown in Figure 7. The history length was  $k = 4$  and  $p = 4$  bins were used to discretize each series, the maximum practical values that could be used following the method of Marschinski and Kantz (see above). Mutual information between the instantaneous velocities of

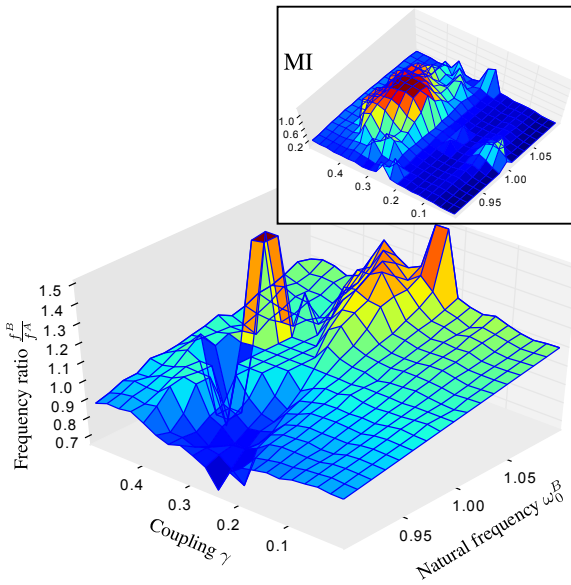


Figure 6: Ratio of peak-to-peak oscillation frequency for  $x_1^A$  and  $x_1^B$  when the natural frequency of the first oscillator is  $\omega_0^A = 1 \text{ rad s}^{-1}$  and the second oscillator is varied near to that, for increasing internal coupling ( $\gamma$ ) in both oscillators. The system parameters were  $Q = 0.67$ ,  $\alpha = 0.1$ ,  $\zeta = 0.3$ ,  $\gamma_c = 10$ ,  $t_{tr} = 400 \text{ s}$  and  $t_1 = 15000 \text{ s}$  with measurement interval  $\Delta t = \frac{1}{20} \text{ s}$ . Inset: mutual information between time series over the same region in parameter space shows the synchronization region more clearly.

the two oscillators (measured by the values of  $\dot{x}_1$  produced by the simulation) is used to measure synchronization, with high values of mutual information implying that the oscillators vary at related speeds and therefore are synchronized. The same binning approach as for transfer entropy is used, but with  $p = 5$  bins. The mutual information is also rendered in the inset plot in Figure 6, which shows that high mutual information corresponds to the frequency locking region.

The relationship between transfer entropy and synchronization is complex. There appears to be a main frequency locking region near  $\omega_0^B = 1 \text{ rad s}^{-1}$  in Figure 7a (where the natural frequencies are most similar) and smaller peaks at larger frequency differences, which are hypothesized to be at points where harmonic resonance along the body allows for greater synchronization between the oscillators. Note that at the mutual information (synchronization) peaks, there is usually a trough in the transfer entropy rate, especially in the approximate range  $1 < \omega_0^B < 1.05$ . Here the synchronization is strongest, and the transfer entropy is not seen because the two systems are coordinated in a highly synergistic manner, such that the coupling appears to be rigid to an outside observer. Because the systems are not generating entropy independently (i.e. the entropy rate  $H(X_{t+1}|X_t^{(k)})$  for either system is 0), no transfer entropy can be measured.

To investigate the notion that the transfer entropy measures directed causal information, the internal coupling  $\gamma$  was set to zero for one of the oscillators (A or B) at a time (with the other retained at  $\gamma = 0.35$ ). Recall that the internal coupling regulates the strength of the signal from the spring extension that is incorporated in the feedback path of the oscillator circuit. Therefore setting the internal coupling to zero for oscillator A will mean that system B cannot have a causal effect on system A, and  $ST_{B \rightarrow A}$  (the transfer entropy from B to A) should be zero, as shown in Figure 7b. Likewise removing the internal coupling from system B results in  $ST_{A \rightarrow B} = 0$  (Figure 7c). In the coupled direction, transfer entropy is generally present. The transfer entropy does not drop to zero in the most synchronized areas of Figures 7b and 7c as it does in the mutually coupled scenario. This suggests that the synchronization is weaker and intermittent, allowing the influence of one oscillator on the other to be measured.

## Conclusions

Transfer entropy from source A to target B is (in a mathematical sense) a Bayesian information gain in moving to posterior knowledge of A from some prior knowledge of B; to infer causality (i.e. “A causes B”) we must be sure that the prior includes *all other causal influences on B that may be correlated with A*, particularly the complete history of B (cf. Lizier and Prokopenko, 2010; Ay and Polani, 2008). Properly measured, transfer entropy will be zero if A does not generate information independently of B.

The above has shown that systems that are weakly synchronizing are capable of this independent information generation, and thus observational transfer entropy can be measured. Furthermore, transfer entropy is *only* found in the case of weak synchrony, and not for systems that are either not truly synchronizing (such as a single mass spring damper coupled to a single driving oscillator), or too rigidly synchronized (as in the case of two very tightly coupled oscillators). Importantly, this means that the observational transfer entropy is not a direct measure of the “strength” of synchrony or causal relationship, because the strongest relationships may show no transfer entropy.

There is a persistent asymmetry in the plots in Figure 7 – in the fully coupled scenario, the transfer entropy appears to be generally higher in the direction leaving the oscillator with higher  $\omega_0$  (remember that  $\omega_0^A$  is always 1, thus in the left hand half of Figure 7a  $\omega_0^A > \omega_0^B$  and notice that generally  $ST_{A \rightarrow B} > ST_{B \rightarrow A}$ ). When feedback coupling is removed in one oscillator, synchronization appears to happen over a larger region when that oscillator has a higher natural frequency, as shown by the asymmetry in the mutual information curves (Figures 7b and 7c). This relation suggests it may be possible to use the transfer entropy to make useful predictions about the consequences of further interventions, with the important caveat noted above that it cannot be a

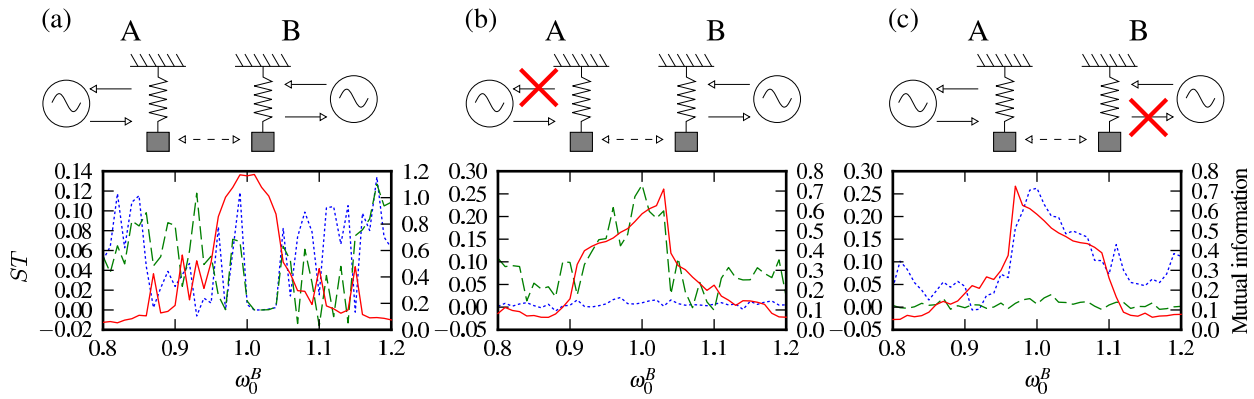


Figure 7: Frequency mutual information (red solid line),  $ST_{B \rightarrow A}$  (blue dotted line) and  $ST_{A \rightarrow B}$  (green dashed line) for double oscillator system, with oscillator  $A$  at  $\omega_0^A = 1 \text{ rad s}^{-1}$  and  $B$  at nearby frequencies as shown. Coupling is: (a) mutual,  $\gamma = 0.35$  in both systems; (b) no feedback in system  $A$  ( $\gamma^A = 0$ ); (c) no feedback in system  $B$  ( $\gamma^B = 0$ ). Other parameters as Figure 6.

perfect method of inferring causality.

Future work will aim to develop a walking robot using analog oscillator controllers in a similar approach to that of Still et al. (2006), but with electrically independent modular limbs that cannot control the mechanical coupling. It is hoped that it may be possible to guide self-organizing synchronization in the limbs via transfer entropy.

### Acknowledgments

Thanks to Phil Husbands for helpful discussions during the course of this work.

### References

- Ay, N. and Polani, D. (2008). Information flows in causal networks. *Advances in Complex Systems*, 11(01):17.
- Ceguerra, R. V., Lizier, J. T., and Zomaya, A. Y. (2011). Information storage and transfer in the synchronization process in locally-connected networks. In *Proc. 2011 IEEE Symposium on Artificial Life*.
- Collins, J. and Stewart, I. (1993). Coupled nonlinear oscillators and the symmetries of animal gaits. *Journal of Nonlinear Science*, 3(1):349–392.
- Der, R., Güttler, F., and Ay, N. (2008). Predictive information and emergent cooperativity in a chain of mobile robots. In *Proc. Alife XI*. MIT Press.
- Hindmarsh, A. (1983). *ODEPACK, a systematized collection of ODE solvers*, pages 55–64. North-Holland, Amsterdam.
- Kiers, K., Schmidt, D., and Sprott, J. C. (2004). Precision measurements of a simple chaotic circuit. *American Journal of Physics*, 72(4):503.
- Klyubin, A., Polani, D., and Nehaniv, C. (2005). Empowerment: A Universal Agent-Centric Measure of Control. In *2005 IEEE Congress on Evolutionary Computation*, pages 128–135.
- Lizier, J. T. and Prokopenko, M. (2010). Differentiating information transfer and causal effect. *The European Physical Journal B*, 73(4):605–615.
- Lungarella, M. and Sporns, O. (2006). Mapping information flow in sensorimotor networks. *PLoS Computational Biology*, 2(10):e144.
- Marschinski, R. and Kantz, H. (2002). Analysing the information flow between financial time series. *The European Physical Journal B*, 30(2):275–281.
- McGeer, T. (1990). Passive Dynamic Walking. *The International Journal of Robotics Research*, 9(2):62–82.
- Pearl, J. (2009). *Causality*. Cambridge University Press, 2nd edition.
- Pfeifer, R., Lungarella, M., Sporns, O., and Kuniyoshi, Y. (2007). On the information theoretic implications of embodiment - principles and methods. In Lungarella, M., Iida, F., Bongard, J., and Pfeifer, R., editors, *50 Years of Artificial Intelligence*, volume 4850 of *Lecture Notes in Computer Science*, pages 76–86, Berlin / Heidelberg. Springer.
- Pikovsky, A., Rosenblum, M., and Kurths, J. (2001). *Synchronization: a universal concept in nonlinear sciences*. Cambridge University Press, Cambridge, UK.
- Pitti, A., Lungarella, M., and Kuniyoshi, Y. (2009). Generating spatiotemporal joint torque patterns from dynamical synchronization of distributed pattern generators. *Frontiers in Neuro-robotics*, 3(2).
- Schäfer, C., Rosenblum, M. G., Kurths, J., and Abel, H. H. (1998). Heartbeat synchronized with ventilation. *Nature*, 392(6673):239–240.
- Schreiber, T. (2000). Measuring information transfer. *Physical Review Letters*, 85(2):461–464.
- Sprott, J. C. (2000). Simple chaotic systems and circuits. *American Journal of Physics*, 68(8):758.
- Still, S., Hepp, K., and Douglas, R. J. (2006). Neuromorphic walking gait control. *IEEE Transactions on Neural Networks*, 17(2):496–508.
- Williams, P. and Beer, R. D. (2010). Information Dynamics of Evolved Agents. In *From Animals to Animats 11*, pages 38–49, Berlin / Heidelberg. Springer.

# Many Hands Make Light Work: Group Evolution and the Emergent Division of Labour

Nicholas Tomko<sup>1</sup>, Inman Harvey<sup>1</sup>, Andrew Philippides<sup>1</sup> and Nathaniel Virgo<sup>1</sup>

<sup>1</sup>CCNR, Evolutionary and Adaptive Systems Group, University of Sussex, Brighton UK  
nt79@sussex.ac.uk, inmanh@gmail.com, andrewop@sussex.ac.uk, nathanielvirgo@gmail.com

## Abstract

Most standard genetic and evolutionary algorithms (GAs) are unable to evolve cooperative solutions to problems where there is a division of labour among genetically different component parts. This is because standard GAs evaluate and select all individuals on the same task which leads to genetic convergence within the population. The goal of evolutionary niching methods is to enforce diversity in the population so that this genetic convergence is avoided. One drawback with some of these niching methods is that they require *a priori* knowledge or assumptions about the specific fitness landscape in order to work. Another issue is that many of these niching methods are not set-up to work on cooperative tasks where fitness is only relevant at the group level. In this paper we present the Group GA which is a group based evolutionary algorithm that can evolve cooperative solutions to problems using emergent niching with minimal *a priori* assumptions. We demonstrate this novel GA on an immune system matching task and explain why we think this type of GA has the potential to effectively solve a wide range of problems that would benefit from being solved cooperatively.

## Introduction

In biology, speciation and niching can be broadly described as the evolutionary process by which a single type of biological organism differentiates into multiple “specialised” organisms, that for instance, take advantage of different resources available in a given environment. In some cases niching produces competing species, but niching can also occur within a single species to produce different specialists that work together to solve a given task. An example of this are bacterial colonies, where within any single colony there are groups of different bacteria doing different jobs, all of which are contributing to the collective well being of the colony. In this case the fitness of the colony depends on the collective symbiotic functionality rather than the fitness of any individual bacteria.

Most standard artificial evolutionary and genetic algorithms (GAs) tend to take a very individual centric view of evolution, where the fittest individuals are selected to produce the next generation of individuals. These types of GAs work well on problems with a single global fitness peak,

where each individual can solve the task on its own; but they are unable to find multiple solutions to multi-peaked problems or solve problems cooperatively, where there is a division of labour between population members which requires different genotypes. For a GA to be able to find cooperative solutions to problems, it must have the following characteristics: (1) It must be able to maintain diversity within the population so that niches can form and (2) it must allow for fitness to be evaluated at the group level.

Evolutionary niching methods solve problem (1) by enforcing diversity in standard GAs so that a single population can be split up into  $n$  different niches. One of the issues with some of the more common niching methods is that they require *a priori* knowledge about the specific fitness landscape to work; in particular whether  $n$  is 2 or 5 or some different number. Most of these evolutionary niching methods use either direct or indirect methods to determine the appropriate number of niches. Direct methods include cooperative coevolution where the number of species is set before evolution begins. Indirect methods include fitness sharing and crowding which rely on a pre-set niching (similarity) radius or some sort of similarity calculation in order to get the population to niche. The other problem with these niching methods is that they are tailored for tasks where each individual in the population can solve the task on its own, not for tasks that are best solved symbiotically where fitness can only be calculated at a group level.

In this paper we present a novel genetic algorithm, the Group GA, which niches based on the evaluation of groups of individuals and therefore can be used to solve tasks that require individuals working together doing different jobs. The Group GA has the added benefit of accomplishing this niching with minimal *a priori* knowledge of the fitness landscape and is able to niche without knowing the optimal number of niches or how the different jobs should be shared out. So unlike the more common niching methods it does not require the number of niches to be set ahead of time nor does it require setting any indirect niching parameter such as a similarity or niching radius.

We demonstrate the emergent niching ability of the Group

GA on an artificial immune system matching task. The goal of this task is to evolve a population of antibodies (protecting agents) to match a set of antigens (harmful invaders). Therefore to solve this task the population of antibodies needs to niche so that it contains different individuals that match different antigens. One reason this task was chosen is because the number of peaks in the fitness landscape can be changed by changing the number of antigens that the population of antibodies needs to match. The other reason for choosing this task is that it makes it very easy to determine when niching has occurred.

In the next section we will briefly review some of the common niching methods as well as a few related evolutionary algorithms that can solve problems symbiotically, where there is a division of labour required. Following our literature review we describe the artificial immune system task and the Group GA in detail. We will then show how the Group GA can be used to evolve a population of antibodies to match a set of four antigens, as well as how it can be used to evolve a population of antibodies that adapts to the addition and removal of antigens during evolution. Finally, we compare the Group GA to other evolutionary methods and discuss the types of tasks we feel the Group GA is best suited to solve.

## Literature Review

We start by reviewing the most common niching methods in artificial evolution. The purpose of these niching methods is to stop the population from genetically converging during evolution as happens when using a conventional GA. All of these niching methods below can be classified as explicit niching methods because they either require the number of niches to be set *a priori* or require an indirect method of enforcing diversity in the population.

We will also briefly discuss SANE and the Binomics GA which are two GAs that are set-up to allow implicit niching to evolve symbiotic solutions to problems. Unlike the explicit niching methods, these algorithms attempt to evolve a diverse, niched population emergently using group evaluation. They also differ from the genetically based niching methods in that these GAs do not require that each individual in the population can solve the task on its own.

## Genetically Based Niching Methods

In this section we briefly describe the common genetically based niching methods. These niching methods function based on the assumption that each individual in the population has its own fitness. For a more in depth summary see Dick (2005) and Mahfoud (1995).

**Fitness Sharing and Clearing** Fitness sharing (Goldberg and Richardson, 1987) is a niching method that relies on some distance metric or similarity measure (either genotypic

or phenotypic) between individuals. By using suitable methods to adjust the fitness of any individual according to how many other similar individuals are within some predetermined niche (similarity) radius, there is a tendency for the population to spread out over multiple peaks or niches in the fitness landscape; thus diversity is maintained. Clearing (Petrovski, 1996) is very similar to fitness sharing but, instead of degrading the fitness of individuals within the same similarity radius or subpopulation, it removes the least-fit individuals within the similarity radius from the population. Horn et al. (1994) show that in Learning Classifier System models where fitness is shared amongst cooperating individuals implicit niching can occur.

**Crowding** Crowding was first introduced by De Jong (1975) as a method of removing similar individuals from a population, with the goal of trying to maintain diversity during evolution. Deterministic Crowding (Mahfoud, 1995) is a specific type of crowding that mates two in the population and then replaces the parent that is most similar to the offspring if the offspring is fitter. It is similar to fitness sharing because there needs to be some similarity calculation done between individual, but unlike fitness sharing there is no requirement to pre-specify a similarity radius.

## Demes and Spatially Structured GAs

An alternative to genetically based niching methods are spatially structured GAs; for a good review see Dick (2005). In these types of GAs, the population is structured within some local geographical distribution (demes) that constrains which members of the population are allowed to be selected or be recombined with one another. By structuring the population into demes more genetic diversity can be maintained across sub-populations.

**Cooperative Coevolution** Cooperative coevolution was first introduced by Potter and De Jong (1994) as a method for function optimisation. In cooperative coevolution the population is pre-divided into different subpopulations, so it can be thought of as a type of spatially structured GA. Each subpopulation represents a subcomponent required to solve the overall task, which means that there needs to be some *a priori* knowledge of the problem so that the appropriate number of subpopulations is chosen. Each subpopulation is evolved separately using a standard GA, but the fitness of the individual members of each subpopulation is based on the performance of the cooperative solutions. In cooperative coevolution speciation is not emergent because the number of subpopulations needs to be determined before evolution begins. For this reason, this class of algorithms has been shown to work well on problems where there is an obvious way of dividing up the population, such as job shop planning and scheduling tasks (Husbands and Mill, 1991; Husbands, 1993; McIlhagga et al., 1996).

## Symbiotic GAs

SANE (Moriarty and Miikkulainen, 1995, 1996), the Binomics GA (Harvey and Tomko, 2010) and simulated ecosystem evolution (Williams and Lenton, 2007) are three examples of GAs that cause implicit niching in the population and attempt to evolve symbiotic solutions to problems. In SANE and the Binomics GA, groups of individuals are evaluated together and then the individuals that are part of the fittest groups are selected to pass on their genes to the next generation. This differs from most standard GAs where individuals are evaluated and then the fittest individuals are selected. These algorithms are relevant to our discussion of speciation/niching because any time a problem is solved symbiotically then implicit niching must be occurring during evolution.

SANE and the Binomics GA have been successfully applied to the evolution of artificial neural networks (ANNs). In both these algorithms the individuals in the population are partial networks that are combined to form fully specified ANNs which are then evaluated. The fitness score of each individual partial network is based on the fitness of the full ANNs that each individual partial network participated in. This means that over time, the individual partial networks that were part of the fittest ANNs will be selected for, while the partial networks that were part of the least fit ANNs will be modified using mutation and recombined with other partial networks. The goal of this method of evolution is to evolve a population of partial networks that symbiotically work together to form high fitness fully specified ANNs.

## The Artificial Immune System Task

We have chosen an artificial immune system matching task to demonstrate the emergent niching abilities of the Group GA. In this section we will describe the details of this task and then in the next section we will describe the Group GA. This task which has previously been used by Forrest et al. (1993) and Potter and De Jong (2000) was chosen because it can be solved cooperatively and clearly illustrates how the Group GA can lead to emergent niching and how it can adapt to a changing fitness landscape, neither of which is possible with a conventional GA. Forrest et al. (1993) used the task to study adaptation in the immune system and Potter and De Jong (2000) solved different variations of this task using cooperative coevolution. We will compare the results of these two papers to the Group GA results later in the paper.

The goal of this task is to evolve a population of antibodies to protect the body from a set of antigens. Very simply speaking, antigens can be thought of as bacteria, viruses or other pathogens and the antibodies can be thought of as the body-guards who mark these antigens for removal. Antibodies in natural immune systems need to be adaptive so that they can combat new and different antigens that enter the body. Therefore this task tries to mimic this challenge of natural immune systems on a very basic level by attempt-

ing to evolve a population of artificial antibodies to match a variable set of antigens.

In this task both the antibodies and antigens are modeled as bit strings. How well an antibody combats a specific antigen is calculated as the number of bit matches between antibody and antigen. For example a [1 0 1 1] antibody matches a [0 0 1 0] antigen at location two and three and therefore the antibody's fitness is equal to two when matched to this antigen. For our purposes the higher the match (fitness) score the better.

Assuming that the length of the antibodies and antigens is the same, when there is more than one antigen in the antigen set the task can be thought of as symbiotic, because it is impossible for a single antibody to match an entire set of antigens on its own. In this case, the population of antibodies needs to evolve in such a way so that it contains specialists to combat each different antigen. Obviously the more antigens there are, the more difficult the task becomes, because the antibody population needs to evolve and maintain a larger number of specialists.

## The Group GA

The Group GA is a novel evolutionary algorithm presented in this paper for the first time. It is based on the Microbial GA (Harvey, 2011) which is a steady-state GA that uses tournament based selection. The Microbial GA is similar to the more familiar GAs, but is minimalist in the sense that it strips away as much as possible, whilst still maintaining the essential components of natural selection which are heredity, variation and selection.

We will first describe the Group GA in general terms and then describe it in terms of the artificial immune system task we present in this paper. What differentiates the Group GA from more conventional GAs is that groups of population members, of some fixed size that is a parameter of the GA (rather than individual population members as in conventional GAs) are evaluated and then selected based on the overall fitness of the group. In other words, the driver of fitness based selection is the relative fitness of an entire group of population members that work together as a unit to solve some task. A single cycle (tournament) of the Group GA can be broken-up into the five following steps:

1. Randomly choose two possibly intersecting groups of population members from the population without regards to fitness.
2. Calculate and assign a fitness score to each group of population members based on the groups' performance on a given task. Fitness is assigned on the group level only; there need not be any way to define or calculate an individual's contribution to the group's fitness score.
3. All members of the group with the lower fitness score are removed from the population and replaced with mutated



copies of the members of the fitter group.

4. The members of the fitter group are put back in the population unchanged.
5. This process is repeated until some pre-defined stopping condition is met.

When we apply the Group GA to the immune system task, the fitness of a group of antibodies is calculated as the average of the best match scores achieved against all the antigens in the set. In other words, to evaluate a group of antibodies, all the antibodies in the group are matched against every antigen in the set and the average of the highest match scores against each antigen is the group fitness. This means that to get a perfect fitness score there has to be at least one antibody that matches each antigen perfectly in the group.

A single cycle (tournament) of the Group GA can be broken-up into the five following steps when applied to the immune system task described in the previous section (see figure 1).

1. Randomly choose two groups of antibodies from the population without regard for fitness
2. Calculate the match scores between all the antigens in the set and each of the antibodies in each group
3. Each group as a whole is assigned a fitness score which is calculated as described above.
4. The group with the lower fitness score is replaced with mutated copies of the antibodies of the more fit group
5. Both groups of antibodies are put back into the population and this process is repeated

We have set up this simulation in such a way that groups of antibodies are randomly chosen from the population and then assigned a fitness based on the ability of this group to match the different antigens in the antigen set. We understand that because an individual antibody can always be assigned its own fitness, some of the genetically based niching methods we reviewed earlier would be able to solve this task without any type of group evaluation. The reason we have used this task to demonstrate the Group GA is because as we will see in the next section it clearly shows how the Group GA causes emergent niching using group evaluation.

The Group GA can be applied more generally to tasks where individual fitness is meaningless because the Group GA randomly selects two groups of population members and uses them to construct two higher level entities that are evaluated and assigned a fitness score. The less fit group of population members is killed off and replaced with a mutated copy of the fitter group. These two groups are then put back into the population and this cycle is repeated. It is important to reiterate that in the Group GA it is the fitness of the group

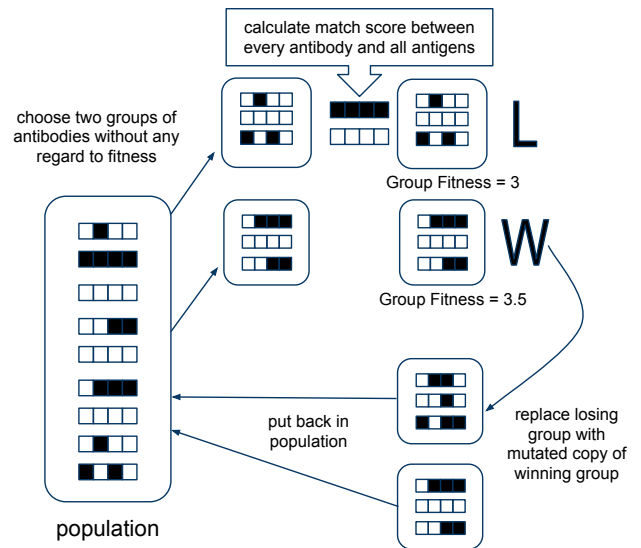


Figure 1: The Group GA as applied to a immune system task with 2, 4-bit antigens

of population members that drives evolution, which is different from most conventional GAs where it is the fitnesses of the individual population members that matters. How fitness is calculated depends on what type of problem is being solved, but regardless it is only the group fitness that matters when determining the tournament winner and loser.

## Evolving Antibodies using the Group GA

In this section we will show how, using the Group GA, a randomly initialised population of antibodies can be evolved to match a set of antigens. In the first experiment we will evolve a population of antibodies to match a fixed set of four different antigens. This is equivalent to the Group GA solving a four-peaked fitness landscape. Then in the second experiment we will evolve a population of antibodies to match a variable set of antigens, where antigens are added and removed during evolution. This second experiment simulates a task where the number of fitness peaks changes during evolution.

In these experiments the antigen and antibodies were 64-bit binary strings. The antibody population size was 100 and the number of antibodies per group was 10. The mutation rate was set to 0.1/64, meaning that at each allele there was a probability of 1/640 of flipping that bit.

Figure 2 shows the antibody population after being evolved for 20,000 tournaments on a four antigen task. The four antigens used in this experiment were: [...0 0 0 0...], [...1 1 1 1...], [1 0 0 0...], and [...1 0 1 0 ...], where these 4-bit patterns are repeated 16 times to make the four full 64-bit antigens. These specific antigens were chosen to try to make the task as difficult as possible. The lower part of figure 2



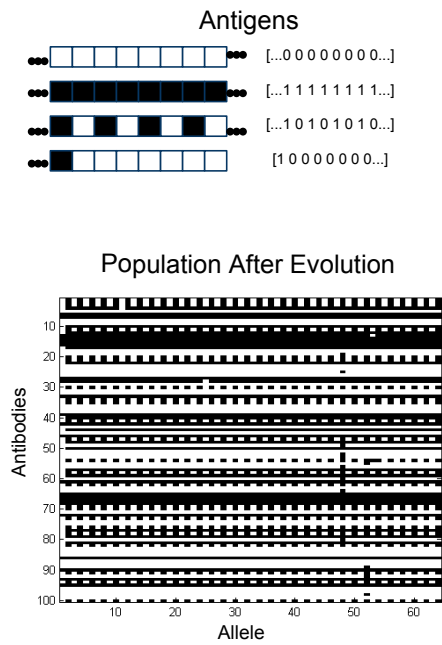


Figure 2: The antibody population after evolution on a 4 antigen task.

(as with the similar plots in later figures) displays each binary genotype in the population horizontally above the next genotype, with white and black representing 0 and 1 alleles respectively. Figure 2 clearly shows how the antibody population has niched during evolution to contains antibodies that perfectly match all four antigens in the set.

Figure 3 is a fitness versus time plot for a single typical run of the four antigen task. The black line shows the group fitness of the tournament winning group of antibodies at each tournament, calculated as described above and the gray line shows the number of antigens covered perfectly by at least one antibody at each tournament. The number of perfect antigens matched perfectly by at least one antibody can range from zero to the total number of antigens in the set. We believe that this is an important measure of performance for this task because if you think of the goal of the antibodies in terms of protecting a body from invasion, then it is important that the population contains at least one antibody to match each antigen. In this figure you can see that throughout evolution the group fitness drops significantly for a tournament or two without decreasing the fitness of the population (number of perfect antibody types). This is because antibody groups are randomly chosen from the population so there is always a chance that a very unfit group is chosen.

Figure 4 shows how the antibody population adapts when antigens are added and removed during evolution. In this experiment, the antigen set initially contained only two antigens [...0 0 0 0...] and [...1 1 1 1...]. At tournament 20K

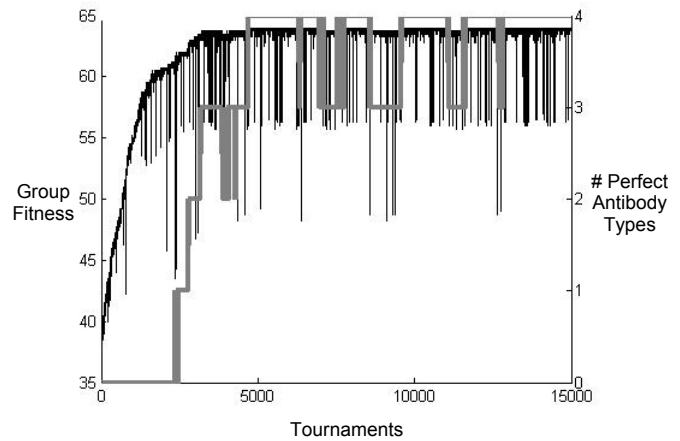


Figure 3: A plot of group fitness (black line) and number of antigens covered perfectly by at least one antibody (gray line) in the population over time for a single typical run of the 4 antigen task.

a third antigen [...1 0 1 0...] was added and evolution was resumed. At tournament 40K evolution was paused again and the [...1 1 1 1...] antigen was removed from the set before evolution was restarted. This figure clearly shows that when the antibody population is evolved using the Group GA the population can adapt to changes in the antigen set, adding and removing different types of antibodies as appropriate. Figure 5 shows the fitness versus time plot for this a single typical run of this task, where antigens are added and removed during evolution. As this figure shows, when an antigen is added, the fitness of the population drops before quickly recovering as the population adapts to match this new invader <sup>1</sup>.

## Comparison to Other Methods

To get a feel for how well the Group GA is able to solve on this task we compared it to both the Microbial GA (Harvey, 2011) and the Binomics GA (Harvey and Tomko, 2010) on the 4 antigen task described above. Using the Microbial GA to solve this task is equivalent to solving it using any standard GA where the fittest individual antibodies are selected. As expected, when we ran the Microbial GA for 100 runs, each run the antibody population converged to match a single antigen in the antigen set, failing to match the other three.

A more interesting comparison is between the Group GA and the Binomics GA. We chose to compare the Binomics GA as opposed to a genetic based niching method such as fitness sharing or crowding because like the Group GA, the

<sup>1</sup>There are potential similarities between the adaptive mechanism of the Group GA and clonal selection that need to be investigated further.

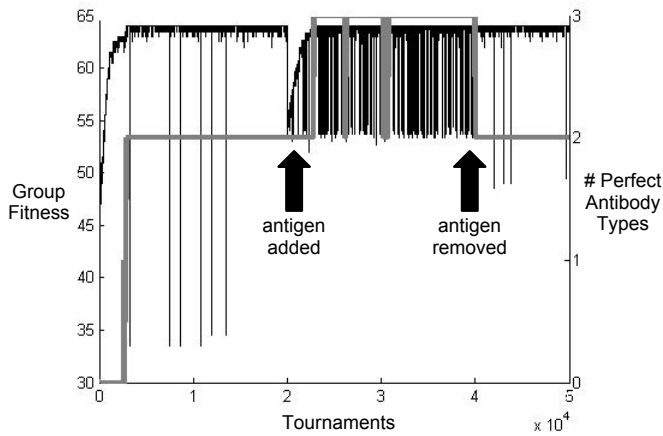


Figure 5: A plot of group fitness (black line) and number of antigens covered perfectly by at least one antibody (gray line) in the population over time for a single typical run of the task where antigens are added and removed during evolution.

Binomics GA was developed to solve cooperative tasks using emergent niching where group fitness is the driver for selection. As applied to this immune system task, the Binomics GA works as follows:

1. Randomly choose two antibodies from the population and compare their stored fitnesses.
2. The antibody with the lower fitness is genetically changed using mutation and recombination.
3. This modified antibody is combined with a group of randomly chosen antibodies from the population.
4. All the antigens are matched against all the antibodies in the group.
5. The fitness of this group of antibodies is equal to the mean maximum match score in the group.
6. All antibodies in the group have their current fitness updated using some sort of time smoothing that takes into account both their historical and newly calculated fitness.
7. All individuals are put back in the population and this cycle is repeated.

The difference between the Group GA and the Binomics GA is that in the Group GA, groups of antibodies are being both evaluated and selected, while in the Binomics GA groups of antibodies are being evaluated, but it is individual antibodies that are being selected based on this group fitness.

Using the same parameters as in the previous experiments, we compared the performance of the Group GA and

the Binomics GA on the 4 antigen task over 10 runs. We decided to compare the performance of these two algorithms based on the number of evaluations it took to evolve a population that contained antibodies that perfectly matched all antigens in the set. Evolution was stopped at 1600 K evaluations if by that point the population did not contain 4 perfect antibodies. Over 10 runs the Group GA took a median number of 278 K evaluations, while the Binomics GA was unable to solve the task within the maximum number of evaluations allowed in any of the 10 runs. It should be mentioned that if the Binomics GA was allowed to run for more evaluations, it was able to niche to match the four different antigens, but nowhere near as efficiently as the Group GA. In the next section we will discuss why we think the Group GA outperforms the Binomics GA to this extent.

## Discussion

In this paper we have presented a novel evolutionary algorithm that can cooperatively solve problems using emergent niching, where fitness is evaluated at the group level. We demonstrated this by using the Group GA to solve a multi-peaked artificial immune system matching task. Our results show that by evolving a population of antibodies using the Group GA, the population niches to match multiple antigens. We have also shown that when antigens are added and removed during evolution, the Group GA allows the antibody population to adapt to this change matching new antigens that are presented.

In the previous section we compared the performance of the Group GA to the Microbial GA and the Binomics GA. Unsurprisingly, the Microbial GA, where individual antibodies are evaluated and selected was unable to solve the multi-antigen task and ended up converging to match a single antigen every run. The Binomics GA, where groups of antibodies are evaluated and individual antibodies are selected, fared much better and was able to niche to match the different antigens, but took a lot longer as compared to the Group GA. We believe that the reason why the Group GA outperforms the Binomics GA methods on this task is related to the difference between what is being evaluated and what is being selected. Studying the subtle differences between evaluation and selection and how varying what is evaluated and selected affects artificial evolution is not part of the scope of this paper, but will be one of the focuses of our future research.

The two key characteristics of the Group GA that differentiate it from the niching methods described in the literature review are: (1) Niching is accomplished emergently without having to know the appropriate number of niches ahead of time or pre-setting any parameter such as a niche radius and (2) fitness is evaluated at a group level which means that the Group GA can be used to solve symbiotic task where fitness is meaningless at the individual level.

For example, this same immune system task was solved

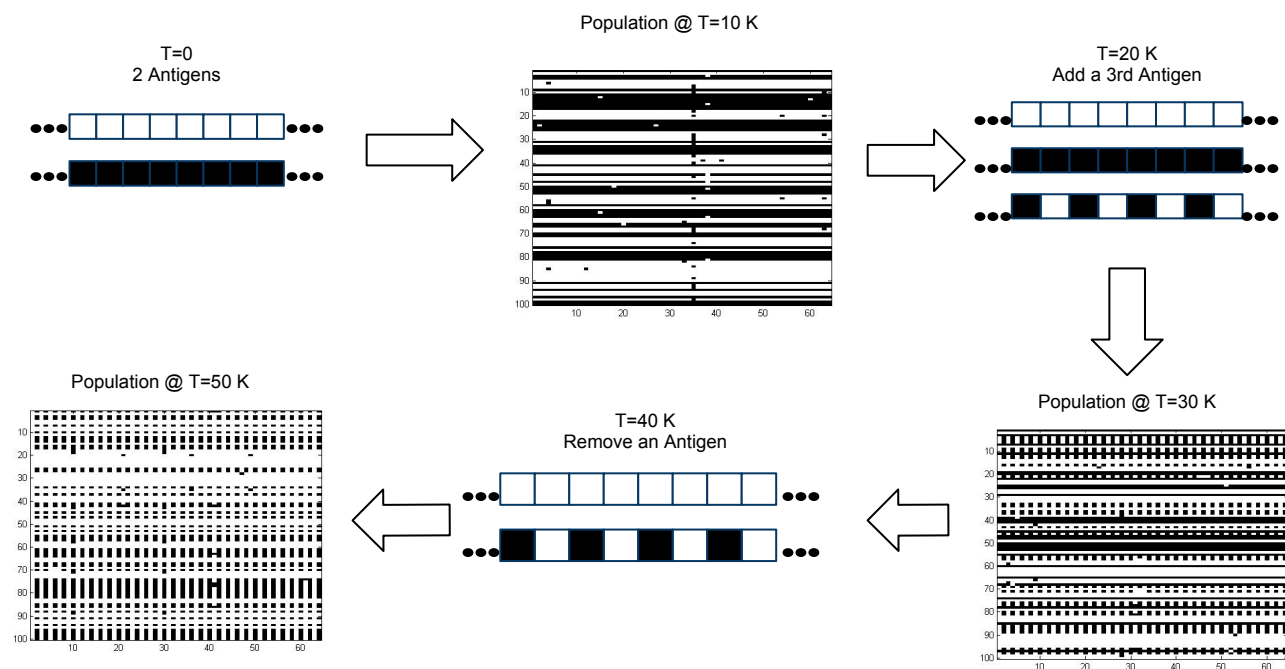


Figure 4: This figure shows how the antibody population adapts during evolution when 64 bit antigens are added and removed (T=10 K corresponds to tournament 10,000).

by Potter and De Jong (2000) using cooperative coevolution where the population was subdivided into  $n$  different species before evolution was started. This method was successful at evolving a population of antibodies to match different antigens as long as the number of different antigens was known *a priori* and the number of antigens remained constant throughout evolution. To overcome these limitations of cooperative coevolution, Potter and De Jong (2000) applied an evolutionary stagnation measure to determine when a new sub-population should be added. This allows antibody species to be added and removed during evolution in response to new antigens, but as Potter and De Jong (2000) state, the level of stagnation at which species should be added or destroyed is task dependent.

This task was also solved by Forrest et al. (1993) using a GA with a best-match fitness scoring scheme. In their algorithm, an antigen is chosen at random and matched against a group of antibodies from the population. Only the antibody in the group with the highest match score gets its fitness increased by its match score, the fitness of all other antibodies remains unchanged. This fitness evaluation step is repeated many times and then the population is evolved using a standard GA. Like the Group GA, this method allows the antibody population to *niche* to match a set of antigens without needing to know *a priori* how many antigens are present. The major difference between this method and the Group GA is that this best match method requires that the fitness

of individual population members can be evaluated on their own. This is possible for this task because each individual antigen can be evaluated on its own by matching it against a single antigen, but tasks where fitness can only be evaluated at the collective, group level will not be able to be solved using this best-match method. In general, the genetically based niching methods described earlier will struggle with this type of symbiotic task where individual fitness is meaningless. An example of this type of task is the evolution of artificial neural networks (ANN) task where the population is made up of partial sub-networks which have no fitness except when they are combined with other sub-networks to form a fully specified networks. Both SANE and the Binomics GA discussed earlier have been used to solve ANN tasks in this way.

For the reasons given above we believe that the Group GA has the potential to be a useful algorithm that can use emergent niching to solve problems where the optimal division of labour is unknown. Going forward, we plan on testing the Group GA on a wide variety of tasks which may benefit from being solved cooperatively in order to find out when it performs well and under what circumstances it performs poorly. We also plan on studying the effect of varying the group size parameter on this immune system task as well as other tasks. Testing the Group GA on an ANN task may be a logical next step, as neural networks can be viewed as a group of neurons symbiotically working together to solve a

problem. We think that the Group GA could be the catalyst for the development of a new class of GAs that specialise in solving tasks cooperatively where there is limited *a priori* knowledge of the fitness landscape.

## References

- De Jong, K. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. Ph.d. thesis, University of Michigan, Ann Arbor.
- Dick, G. (2005). A comparison of localised and global niching methods. In *17th Annual Colloquium of the Spatial Information Research Centre (SIRC 2005: A Spatio-temporal Workshop)*, pages 91–101, Dunedin, New Zealand. Citeseer.
- Forrest, S., Javornik, B., Smith, R. E., and Perelson, A. S. (1993). Using Genetic Algorithms to Explore Pattern Recognition in the Immune System. *Evolutionary Computation*, 1(3):191–211.
- Goldberg, D. and Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimisation. In Grefenstette, J., editor, *Proc. of the Second International Conference on Genetic Algorithms*, pages 41–49, Hillsdale, NJ. Lawrence Erlbaum Associates.
- Harvey, I. (2011). The Microbial Genetic Algorithm. In Kampis, G., Karsai, E., and Szathmary, E., editors, *ECAL 2009, Part II. LNCS 5778*, pages 126–133, Heidelberg. Springer.
- Harvey, I. and Tomko, N. (2010). Binomics : Where Metagenomics meets the Binary World. In *Proceedings of Artificial Life XII, 12th Intl. Conf. on the Synthesis and Simulation of Living Systems*, Odense, Denmark.
- Horn, J., Goldberg, D., and Deb, K. (1994). Implicit niching in a learning classifier system: Nature’s way. *Evolutionary Computation*, 2(1):37–66.
- Husbands, P. (1993). An ecosystems model for integrated production planning. *International Journal of Computer Integrated Manufacturing*, 6(1):74–86.
- Husbands, P. and Mill, F. (1991). Simulated co-evolution as the mechanism for emergent planning and scheduling. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 264–270. Morgan Kaufmann Publishers.
- Mahfoud, S. W. (1995). *Niching Methods for Genetic Algorithms*. PhD thesis, University of Illinois at Urbana-Champaign.
- McIlhagga, M., Husbands, P., and Ives, R. (1996). A comparison of optimization techniques for integrated manufacturing planning and scheduling. *Parallel Problem Solving from Nature IV*, pages 604–613.
- Moriarty, D. and Miikkulainen, R. (1995). Learning Sequential Decision Tasks. In Honavar, V., Patel, M., and Balakrishnan, K., editors, *Advances in the Evolutionary Synthesis of Neural Systems*, Cambridge, MA. MIT Press.
- Moriarty, D. E. and Miikkulainen, R. (1996). Efficient reinforcement learning through symbiotic evolution. *Machine Learning*, 22:11–32.
- Petrowski, A. (1996). A clearing procedure as a niching method for genetic algorithms. *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 798–803.
- Potter, M. and De Jong, K. (1994). A cooperative coevolutionary approach to function optimization. In Davidor, Y. and Schwefel, H., editors, *Parallel Problem Solving from Nature (PPSN III)*, pages 249–257. Springer Verlag.
- Potter, M. and De Jong, K. (2000). Cooperative coevolution: an architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29.
- Williams, H. and Lenton, T. (2007). Artificial ecosystem selection for evolutionary optimisation. In Almeida E Costa, F. E. A., editor, *Advances in Artificial Life: Proceedings of the 9th European Conference on Artificial Life*, pages 93–102, Berlin. Springer Verlag.

# An interactive wall game as an evolution of proto language

Ryoko Uno,<sup>1</sup> Keisuke Suzuki<sup>2</sup> and Takashi Ikegami<sup>3</sup>

<sup>1</sup>Division of Language and Culture Studies, Institute of Technology, Tokyo University of Agriculture and Technology

<sup>2</sup>Laboratory for Adaptive Intelligence, RIKEN, Brain Science Institute

<sup>3</sup>The Graduate School of Arts and Sciences, The University of Tokyo  
ryokouno@cc.tuat.ac.jp, ksk@brain.riken.jp, ikeg@sacral.c.u-tokyo.ac.jp

## Abstract

A new interactive "wall game" is proposed in which two human players alternatively configure a pattern to communicate. A pattern consists of 3x3 sites, on which a player can place one of three symbols. The two major findings in this paper are i) the subjects mainly communicated in two modes. Either the subjects changed the pattern by watching the pattern as it is (dynamical mode) or by having narrative reflection (metaphorical mode). ii) Subjects switched between these two modes. Most of the experiments in evolutionary linguistics are based on "task-oriented communication" and they observe the emergence of lexical items. In contrast, our experiment explores whether "communication without purpose" leads to the emergence of complex rules such as linguistic grammar. We argue that the switching between the two modes observed in our experiment can be seen as a grammatical process in the sense that it is a procedure to take an internal state outside using the media (i.e., patterns in the wall game). Under this hypothesis, the players' exploration of the media becomes a crucial step in the emergence of language and grammar.

## 1. Introduction

Artificial life studies provide a test bed for exploring how symbols and grammars emerge in minimally interacting systems through computer simulation. For the last 10-15 years, artificial life studies have contributed greatly to this direction, and the origin and evolution of language has become a target of many scientific studies (see e.g. Steels, 1996, 2005; Hashimoto and Ikegami, 1996; Rizzolatti and Arbib, 1998; Vogt, 1998; Cangelosi and Harnad, 2000; Sugita and Tani, 2005 etc.). For example, Steels and Kaplan (2001) have developed a platform for studying the interaction between two artificial agents acting as speaker or hearer. In this approach, a population of robots develops a shared vocabulary and a corresponding ontology while playing language games (i.e., ritualized social interactions that follow a specific script).

More recently, there are many researches based on experiments using human subjects (e.g., Steels, 2006; Selten and Warglein, 2007; Scott-Phillips and Kirby, 2010) as a new approach to the origin of language. Subjects communicate through a communication tool and some structured system emerges. Some of these studies testify to a hypothesis that is

raised by computational simulation studies. For example, the "iterated learning model," which is a model of vertical and horizontal cultural transmission, was proposed by Kirby (2002). It was originally studied as a computational model and later the model has been adjusted to experiments using humans (Kirby, Cornish and Smith, 2008).

Among many studies of "language evolution in the laboratory," Galantucci (2005) introduces one of the most influential experiments. In his experiment, two subjects who are staying in different rooms play a video game together over a monitor. They have to be cooperative to get a high score. They are allowed to communicate using a special communication tool. This tool allows the subjects to draw graphics but not letters. As the experiment proceeds, the difficulty of the video game increases. The pairs that ended the game with success shared many signs for rooms and enemy, which were drawn with the communication tool.

In most of the researches adopting an experimental approach, the final outcome often consists of lexical items. This comes from the fact that in most of the experiments, subjects are asked to perform a task together to make them communicate with each other.

Not only the lexicon but grammar is an integral part of a linguistic system. To get more variations in results, the communication observed in the experiments should not be limited to those that are task oriented. For example, we assume that "communication without purpose" can be important to trigger proto-language with both grammar and lexicon in experiments. This idea is supported by the research in developmental psychology: infants are known to be engaged in two types of proto-linguistic communication. The first is communication with an aim, such those that are task oriented. The second is a communication without an aim, in other words communication whose aim is communication itself (Bates, 1976). Gómez, Sarria and Tamarit. (1993) argue the importance of the second type of communication. It is pointed out that the ability to communicate without purpose is an indicator for the ability called "theory of mind." With "theory of mind," one can infer other people's minds, which are different from one's own. And this ability is known to be integral to the use of language grammar properly (Tager-Flusberg, 1993). Tomasello (2003) has also argued the

importance of shared attention in development and pointed out the role of communication just to share communication in acquiring language.

Actually, Uno, Marocco, Nolfi and Ikegami (in press) made an attempt to use the A-life approach to explore the relationship between communication without purpose and the emergence of grammar. The agents were supposed to stay together in the target area using signals. However, when agents were given uncertain information regarding the target area, they start staying together outside the target area using newly created signals, which was argued to be a proto-declarative sentence: a sentence used to share intentionality.

In this paper, we are going to take an experimental approach to see how communication gets structured when there is no purpose. We explore what are the characteristics of human communication (which might possibly be implemented in artificial systems) when individuals are just having fun. We asked subjects to communicate using our communication tool, which is called a “wall game.” The results show that there are two modes of communication. What emerged from the subjects’ communication is not a set of lexical items but the way an internal state of mind can be expressed as an external message. We argue that this can be seen as a proto-grammar.

Section 2 explains the basic design of our experiment. Section 3 and 4 show the results of the experiments. Finally, section 5 analyzes and discusses the results of the experiment.

## 2. Description of the Experiment

Twenty-six subjects (13 pairs) were asked to communicate using an artificial communication system, where the expressions were the spatial pattern of the triplet in a 3-by-3 bit square. They were allowed to rewrite the pattern alternatively. We call this pattern a “message.”

For the first 9 pairs, each subject sent 8 messages in turn, which is 16 messages in sum. For the next 4 pairs, each subject sent 15 messages and 30 in all. After all messages were exchanged, we asked them to report their intentions behind the sent messages, and their interpretations of the received messages in natural language. (Henceforth we call this data the “intention report.”)

We conducted the experiment mainly in Japanese. The reports shown in this paper are translated into English by the authors.

The two subjects stayed in different rooms. The messages were sent to each other over the Internet. Figure 1A shows a screen where one can compose messages. All the messages that are sent and received are shown to the subjects so that they can compose their messages based on their communication history. Figure 1B shows how the history of exchanged messages is displayed to the subjects.

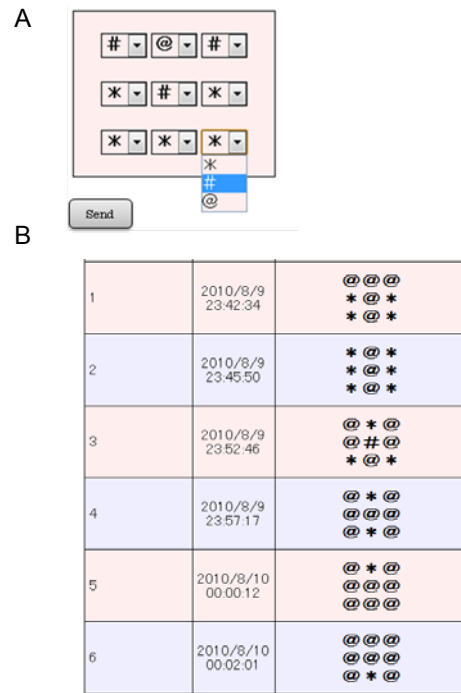


Figure 1: Two screen shots. A is a message composer. B is the timeline of exchanged messages.

Here are some examples of exchanged messages from our data. Player A sent (1) and Player B answered to it with (2). Then Player A replied to it with (3). Finally, (4) is an answer to (3) by Player B.

(1)	(2)	(3)	(4)
From A to B	From B to A	From A to B	From B to A
@ @ @	@ @ @	# # #	# # #
@ * @	@ * #	# # #	# # @
@ @ @	@ @ @	# # #	# @ @

Table1: Exchanged messages between Player A and Player B

We made a linguistic analysis of the intention report and mathematical analysis to the patterns. The results are given in the following sections, 3 and 4.

We performed the experiment under three conditions:

### Condition 1

Messages are exchanged between two subjects. Subjects write intention reports after exchanging all the messages (Appears in Experiment 1, 2, and 3).

### Condition 2

Messages are exchanged between two subjects. Subjects write intention reports in every round and players exchange messages (Appears mainly in Experiment 2).

### Condition 3

Each subject plays with the game on his/her own. Subjects write intention reports after writing all the messages (Appears in Experiment 3).

The basic game is condition 1. In condition 2, the timing of writing the report differs from that in condition 1. In condition 3, the game is played by a single player.

### 3. Experiment 1: Two modes of communication

#### 3.1 Linguistic analysis

To begin with, we analyzed the intention report from a linguistic point of view. We categorized the reports as three types: dynamical report, metaphorical report, and others.

What we call a “dynamical report” is a literal description of the patterns in the messages. For example:

- (D1) All kinds of symbols are used.  
(D2) The pattern is scrolled from left to right.

In these reports, patterns are described just as they are. The messages that these reports are made for are shown in Table 2 below.

On the other hand, in what we call a “metaphorical report,” the subjects create a story based on the symbols inside the pattern. They are not describing the pattern as it is. Instead, they are using metaphors (in the sense of Lakoff [1987]). They describe symbols or a string of symbols as something else. Here are some examples:

- (M1) A rabbit is in a cage.  
(M2) The rabbit made a hole in a cage to escape.

Here the player sees the symbol “\*” as a rabbit and a sequence of “@” as a cage, and “#” is a hole.

In this system, there is no way for one player to transmit her story to the other player. For example, while Player A intended to express a rabbit using the message shown as (M1) in Table 2, Player B made the following intention report for the same pattern:

- (D3) \* is surrounded by @.

In the category called “others,” the reports are not strongly connected to the patterns. For example, we have emphatic expressions such as (O1) or feelings of the players, which are irrelevant with the patterns such as (O2):

- (O1) Hello. Nice to meet you.  
(O2) This experiment is difficult.

(D1)	(D2)	(M1)	(M2)	(O1)	(O2)
* # @	@ # *	@ @ @	@ @ @	@ * @	@ # #
* # @	@ # *	@ * @	@ * #	* @ *	# * #
* # @	@ # *	@ @ @	@ @ @	# # #	* * *

Table 2: Examples of messages three types of reports are made for

The ratio of reports in each category used by each pair is given in Figure 2. It is shown that most of the reports are either dynamical or metaphorical.

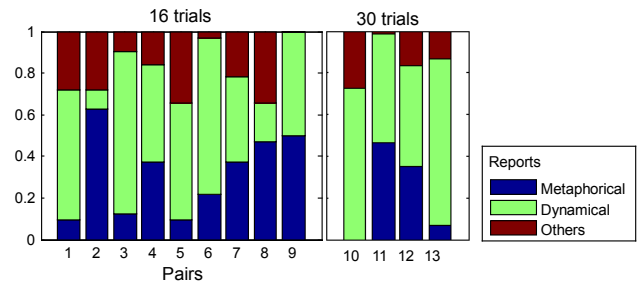


Figure 2: Ratio of report types in Experiment 1. It is calculated from the accumulated reports of two players who exchanged messages. The ratio between metaphorical and dynamical reports varies over pairs.

#### 3.2 Mathematical analysis 1

In order to see the characteristics of the wall patterns in the two different report categories, we calculated the correlation between the Hamming distance of adjacent patterns and the frequency of each type of the reports. Hamming distance is defined as the number of changes required to match one character string with another string. Therefore, we regarded the wall patterns as linear character strings (e.g., (D1) in Table 2 is regarded as “\*#@\*#@\*#@”) to calculate it. The larger the Hamming distance of a couple of patterns, the less they are similar. In order to treat the report under mathematical analysis, the two categories of the report are indexed by counting the number of them in each turn (i.e., the metaphorical index is scored 2 for when both subjects interpret metaphorically, 1 for when one subject does, and 0 for when neither do). We calculated the correlation coefficient between Hamming distances and both the metaphorical indexes, and the dynamical indexes in each turn.

The results are shown in Figure 3. We found that when the Hamming distance between successive patterns gets smaller, the human subjects tend to use metaphorical reports. On the other hand, the Hamming distance between successive patterns gets larger when subjects use dynamical reports.



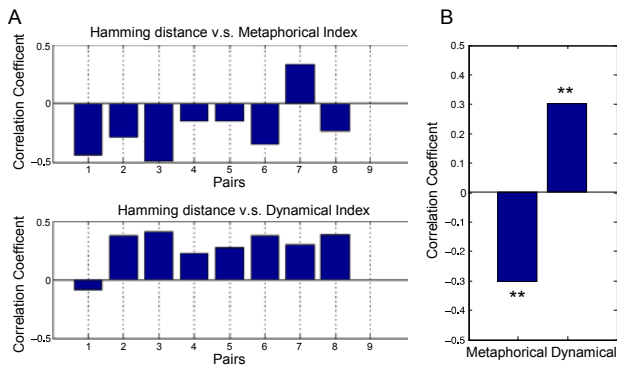


Figure 3: A: Correlation coefficient between Hamming distance and the number of two categories in reports for each pair. B: The same evaluation across all the pairs. Hamming distance has significant ( $p < 0.01$ ) positive correlation with the dynamical report, and significant ( $p < 0.01$ ) negative correlation with the metaphorical report.

### 3.3 Mathematical analysis 2

We also drew a state transition graph between successive patterns. To analyze the transition state of the patterns, 16 messages were not enough to have statistically valid results. We therefore focused on pairs 10 to 13 who exchanged 30 messages (15 each) in one trial. They performed two trials under two different conditions (i.e., conditions 1 and 2. See Section 2 for an explanation). We are going to discuss the contrast between conditions 1 and 2 in detail in Section 4. The point is that under condition 2, more metaphoric reports were tended to be used compared to condition 1.

In order to create the transition graph, we grouped the patterns used in a game by the numbers of symbols the pair used. We first separated the patterns into three rows, and grouped each row using only the constituent ratio of the symbols (e.g., “\*@\*” is grouped into “210”, “###” is grouped into “003”, etc.). Thus, each line represents 1 of 10 groups (0 for “012” ... 9 for “210”). We then assigned the groups a triple-digit number (e.g., 091 for “#@#/\*@\*/@#@”). Finally, we grouped all the patterns used in the game from the number, and calculated the transition between them.

Figure 4A shows the state transition graph calculated for the pair 11. The linearity of the transition graphs is defined as follows: “the number of nodes divided by the number of edges of a graph.” A linearity of the pair 11 under condition 1, whose main communication mode is dynamic, is calculated as 0.89. And the linearity of Pair 11 under condition 2, whose main communication mode is metaphoric, is as 0.97. The analysis of this pair suggests that the metaphoric mode has a tendency of having a lower linearity than the dynamic mode.

Figure 4B shows the correlation coefficient between the linearity and the number of the two report modes in all the trials by 4 pairs. The result shows that same types of transition are used repetitively in the dynamic mode but not in the metaphoric mode.

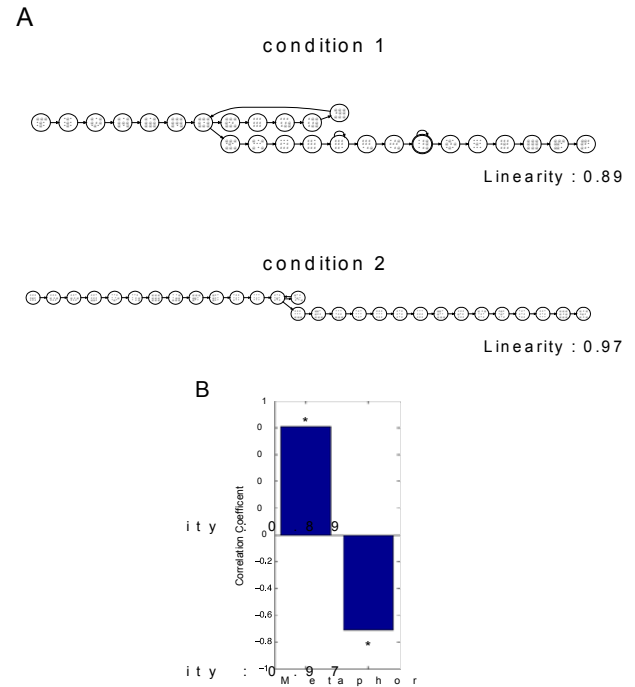


Figure 4:

A: Examples of state transition graphs obtained from results of the pair 11. Here, the linearity is defined as the ratio between the number of nodes and the number of edges. A higher Linearity is observed in condition 2, compared with condition 1.

B: The correlation coefficient between the linearity and the number of the two report modes in the 8 games. A positive correlation can be seen between the linearity and the metaphoric mode, and a negative correlation between the linearity and the dynamic mode.

Here we point out that there are two major modes in communicating with the system introduced in Section 2., metaphoric mode and dynamical mode. During the game, the subjects enjoyed processing patterns and trying to assign meanings to them. In a report, the former shows up as a literal description of dynamic patterns and the latter story is told using metaphors. This difference is correlated with the difference in changing the patterns, which can be partly calculated with the Hamming distance and linearity of the transition states.

## 4. Experiments 2 and 3

To know further about the two modes of communication pointed out in the last section, we made two additional experiments. Below we briefly review each experiment.

#### 4.1 Experiment 2: Message-by-message report

We asked 4 pairs who exchanged 30 messages in Experiment 1 (which we call condition 1) to exchange an additional 30 messages in the new trial (which we call condition 2). This experiment is to make the intentions behind the messages clear and to see the effects on subjects' behavior. In the new experiment, in every round players exchanged the messages they had in order to compile reports of their intentions.

Compare the ratios of the metaphoric and dynamic reports of conditions 1 and 2 shown in Figure 5. This result suggests that when subjects are more conscious of the intention of the message, they tend to be engaged in the metaphoric mode rather than the dynamic mode.

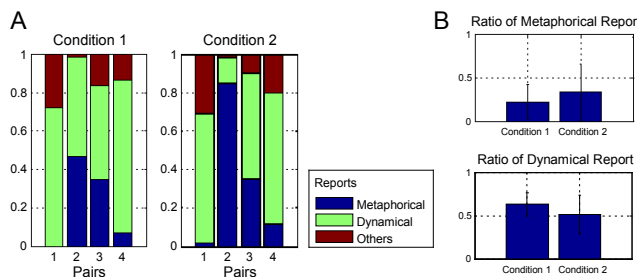


Figure 5:

A: Ratio of reports in each category in condition 1 and 2. In condition 1, the players did not have to make "intention reports" during the game. In condition 2 the players had to make "intention reports" every time they sent or received messages. It is calculated in the same manner as in Figure 1.

B: The ratios of the metaphoric reports and dynamic reports are averaged across all the pairs. There seems to be a tendency for dynamic reports to be more ascendant than metaphoric reports in condition 1, while the opposite tendency can be seen in condition 2.

#### 4.2 Experiment 3: Solitary play

In the third experiment, we asked each subject to play with the game by him/herself (we call this condition 3). We asked one of the subjects who experienced Experiment 1 to make 30 messages by him/herself without having another player "behind the wall" asking him/her to report his/her intentions. Compare condition 1 in Fig. 5 and condition 3 in Figure 6. The result reveals that subjects tend to use either one of the modes, not both of them, when they have no one to communicate with.

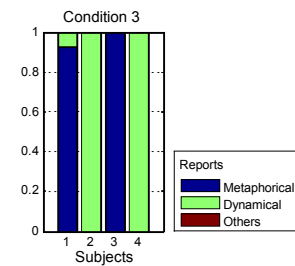


Figure 6: Ratio of reports in each category in condition 3. In this condition, subjects play the wall game alone. The reports are extremely biased into either metaphoric or dynamical.

At least the players get more varieties in behavior when they are together. Let's see an example that is congruent with the results in Experiment 2. Table 3 shows an exchange between two players.

(1)	(2)	(3)	(4)
From A to B	From B to A	From A to B	From B to A
# * *	# # #	* * *	* * @
# @ *	# @ #	* @ @	* * @
# # #	# # #	# # #	# # #

Table 3: Exchanged messages between Player A and Player B

Player A interprets the whole exchange in metaphoric mode. Below are the intention reports by player A for (1) to (4).

- (A1) @ is me and \* is a cherry blossom. Shall I go out by myself?
- (A2) I am also alone.
- (A3) It is more fun if we stay together.
- (A4) Different scene. Here \* and # are people. @ and @ joined them.

On the other hand, player B is communicating with dynamical mode from (1) to (3). At (4) he starts to use the metaphoric mode. Here are the intention reports by player B for (1) to (4).

- (B1) More #s.
- (B2) I added more #s.
- (B3) @ was added. \* appeared again.
- (B4) @ looks like an cute animal. # and \* are environments. So I moved @.

Player B started to change the mode of communication after communicating with player A. As shown in this sample, an interaction between two players facilitates switching between two modes.

All of the experiments that show the characteristics of two modes of communication using our wall game are

summarized in the following table. We interpret the results in the next section.

	Experiment 1		2	3	
	Intention	Hamming	Transition	On-spot	Alone
	Report	Distance	state	report	
Metaphorical mode	Metaphorical	Smaller	More linear	increase	N/A
Dynamical mode	Dynamical (literal)	Larger	Less linear	decrease	N/A

Table 4: Summary of the experimental results

## 5. Analysis and discussio

### 5.1 Interpretation of two modes

Different from previous evolutionary linguistic experiments, players of the wall game were asked to communicate without a purpose. The only motivation is to enjoy communicating with each other.

In this game, the easiest way to compose a message is to mimic what the other player did. However, this strategy has to be avoided because the communication becomes monotonous and predictable so that the players can easily get bored.

Accordingly, we assume that the behavior of the players is the one that tries to avoid mimicking each other and instead they need a strategy to make messages that has novelty for the other player. Two modes of communication can be understood from this point of view.

First of all, the dynamical mode is a mode in which the player pays attention to the patterns in the messages as they are. Therefore, the reports are literal descriptions of the patterns (dynamical report). To make an interesting change in messages only with patterns, there must be a distinct change. This explains why the Hamming distance calculated in Experiment 1 was relatively large. The frequently used patterns that can make interesting transitions are limited. For example, patterns with three lines are often used, as is shown in the transition from (D1) to (D2). This explains the result of Experiment 2, which shows that the transition state of the dynamical mode was less linear, which means that the same pattern was frequently used.

Turning to metaphorical mode, the players make their own stories based on the transition of the patterns. The story itself cannot be transmitted to the other player in this game. So for the Player B who does not share the story, the message by Player A in metaphorical mode is unpredictable and novel. It has been pointed out that metaphor helps people extend their understanding (Lakoff, 1987) and make inferences (Thibodeau and Boroditsky, 2011). In addition, we want to point out that metaphor helps people behave in a creative manner based on the observation in our experiment.

In metaphorical mode, as shown in Experiment 1, the Hamming distance is small. This is understandable when we realize that even small changes can be meaningful in a story. Compare (M1) and (M2). As shown in Experiment 2, the linearity of the transition pattern is big, that is, the same patterns are rarely used. This can be explained by the fact that in metaphorical mode what is meaningful is the difference between the current diagram and the last one. This means that

there is no particular pattern that has to be used in metaphorical mode.

In Experiment 2, we tried to capture the relationship between two modes and “attention”.

The subjects enjoyed processing patterns and trying to assign meanings to them. In a metaphorical mode, subjects conveyed a message more consciously, by paying more attention to the messages. In contrast, in the former process, that is, in dynamical mode, the subjects explored the texture of the 3x3 bits until they became so familiar with the game itself that it became consciously transparent

Let’s move to the result of Experiment 3. It shows that when the players are alone, they tend to use either one of the modes. When two players are together, both modes occur in communication. This suggests that coexistence of the two modes is enhanced by communication.

### 5.2 Proto-linguistic grammar

The outcome of the wall game experiment is two modes and the player’s switching behavior between the two modes. These two modes together form a procedure of taking our inner thoughts and our feelings and then expressing them outside through the media, i.e., in this case, the wall game. What we got is apparently not lexical items but a process: a process that can be seen as a process of producing linguistic expression. It corresponds to grammatical rules, which are used to compose sentences in natural language. The interesting observation here is that the “grammatical process” observed includes the exploration of the media. In dynamical mode, players try to find out the possibility of the pattern, and what kind of patterns can be used to make a distinctive message.

Our hypothesis is that the exploration of the nature of the media is an integral part of the emergence of grammar. Just by looking at natural language, whose media is already transparent to the users, it is difficult to see whether this is true or not. From this perspective, evolutionary linguistic approach seems to be promising. Since the results presented in this paper are still far from proving this hypothesis, currently, as shown in figure 7, we are making wall games with various textures so that how the player explores the media can be observed. This attempt might give us a way to look into all kinds of languages that would be theoretically possible.

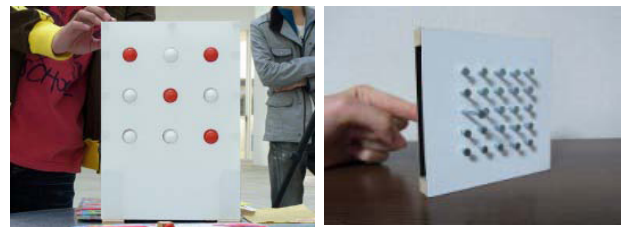


Figure 7: We are now making various types of wall games to analyze the exploratory behavior of players in playing with the wall. (Designed by Seara Ishiyama)

## References

- Bates, E. (1976). *Language and Context: the Acquisition of Pragmatics*. New York: Academic Press.
- Cangelosi A. and Harnad, S. The adaptive advantage of symbolic theft over sensorimotor toil: Grounding language in perceptual categories. *Evolution of Communication*, (4)1: 117-142.
- Galantucci, B. (2005). An experimental study of the emergence of human communication systems. *Cognitive Science* 29:737-767.
- Gómez, J. C., Sarria, E., and Tamarit J. (1993). A comparative approach to early theories of mind: ontogeny, phylogeny and pathology, Baron-Cohen, S. Tager-Flusberg, H. and Cohen, D. J. editors. *Understanding Others Minds: Perspectives from Autism*. Oxford: Oxford University Press. 195-207
- Hashimoto, T. and Ikegami, T. (1996). Emergence of net-grammar in communicating agents, *BioSystems* 38: 1-14.
- Sugita, Y. and Tani, J. (2005). Learning semantic combinatoriality from the interaction between linguistic and behavioral processes, *Adaptive Behavior*, (13)1: 133-152.
- Kirby, S. (2002). Natural language from artificial life, *Artificial Life*, 8(2): 185-215.
- Kirby, S., Cornish, H. and Smith, K. (2008). Cumulative cultural evolution in the laboratory: An experimental approach to the origins of structure in human language. *Proc. Natl. Acad. Sci. U. S. A.* 105. 10681-10686.
- Lakoff, G. and Johnson M. (1980). *Metaphors We Live By*. Chicago: University of Chicago Press.
- Rizzolatti, G., Fadiga, L., Gallese, V. and Fogassi, L. (1996). Premotor cortex and the recognition of motor actions, *Cognitive Brain Research*, 3: 131-141.
- Scott-Phillips, T. C., and Kirby, S. (2010). Language evolution in the laboratory. *Trends in Cognitive Sciences*, 14(9): 411-417.
- Selten, R. and Warglien M. (2007). The emergence of simple languages in an experimental coordination game. *Proc. Natl. Acad. Sci. U. S. A.* 104. 7361-7366.
- Steels, L. (1996) A self-organizing spatial vocabulary. *Artificial Life Journal*, 2 (3): 319-332.
- Steels, L. and Kaplan, F. (2001). AIBO's first words: The social learning of language and meaning. *Evolution of Communication*, (4)1 4-1: 3-32.
- Steels, L. (2005). The emergence and evolution of linguistic structure: from lexical to grammatical communication systems, *Connection Science*, (17)3, 4: 213-230.
- Steels, L. (2006). Experiments on the emergence of human communication. *Trends in Cognitive Sciences*. 10(8): 347-349.
- Tager-Flusberg, H. (2000). Language and understanding minds: connections in autism, Baron-Cohen, S. Tager-Flusberg, H. and Cohen, D. J. editors. *Understanding Others Minds: Perspectives from Autism*. Oxford: Oxford University Press. 124-149.
- Thibodeau, P.H. and Boroditsky L. (2011). Metaphors we think with: The role of metaphor in reasoning. *PLoS ONE* 6(2): e16782. doi:10.1371/journal.pone.0016782
- Tomasello, M. (2003). *Constructing a Language: A Usage-Based Theory of Language Acquisition*. MA: Harvard University Press.
- Uno, R., Marocco, D., Nofi S. and Ikegami, T. (in press, 2011/06) Emergence of protosentences in artificial communicating system. To appear in *IEEE Transactions on Autonomous Mental Development*, 3(2)
- Vogt, P. (1998). Perceptual grounding in robot. *Proceedings of the 6th European Workshop on Learning Robot*, A. Birk and J. Demiris, Eds. Springer-Verlag: 126-141.

# A decomposition theorem in chemical organizations

Tomas Veloz<sup>1</sup>, Bryan Reynaert<sup>2</sup>, David Rojas-Camaggi<sup>2</sup> and Peter Dittrich<sup>3</sup>

<sup>1</sup>Department of Psychology, University of British Columbia

<sup>2</sup>Department of Biology, Universidad de Chile

<sup>3</sup>Friedrich-Schiller-University Jena, Institute of Computer Science, Bio Systems Analysis Group, D-07743 Jena  
tomas.veloz@ubc.ca

## Abstract

The Chemical Organization Theory (COT) is an abstract reaction network model that has a deep connection to autopoiesis as they share the same central topic: Organization. The main characteristic of autopoietic systems is that they preserve their own organization; this constitutes their identity. In terms of COT, organizations are special reaction networks which are closed and self-maintaining. Organizations compose the majority of stable behaviours of a reaction network (Peter and Dittrich, 2011), in particular every fixed point can be mapped to an organization (Dittrich and Di Fenizio, 2007). Obtaining the set of organizations of a network is a central objective in COT, but it is usually a complex computational task. This work intends to reveal the underlying mathematical structure of organizations. We state a theorem of decomposition for organizations to understand the difficulties of verifying if a set of molecular species is an organization. This suggests a step towards the development of more efficient algorithms and the classification of reaction networks in terms of how complex it is to obtain its set of organizations. We also discuss the consequences of this theorem in relation to autopoietic systems.

## Introduction

During a 30-years period, from the 1950's to the 1980's, the field of biological systems and their generalized properties saw the birth of multiple theories (Eigen and Schuster, 1977; Kauffman, 1969; Maturana and Varela, 1973; Wiener, 1948; von Bertalanffy, 1968; Rosen, 1958). A wealth of formalisms were laid out, which focused on different perspectives on the fundamental properties of living systems, but as it was to be expected, there have been deep similarities between most of these theories (Hordijk and Steel, 2004; Jaramillo et al., 2010; Letelier et al., 2003).

Since their conception, most of these theories have been consigned to the theoretical domain having little incidence in applied sciences, with the possible exception of what is currently known as *systems biology*.

This situation may be because the process of translation between the language employed in these theories and the language commonly used in biology is not trivial (Cornish-Bowden et al., 2007). The chemical organization theory, inspired by Fontana and Buss (1994), provides an interesting departure from this tendency as it provides a language

which is not only clear and well-defined but also correlates directly to the biomolecular domain. Due to its mathematical foundations theorems can be formally proven and developed (Benkő et al., 2009; Peter and Dittrich, 2011; Peter et al., 2010). Also, COT is a powerful tool to analyze the asymptotic behaviour of reaction networks that other analytic or simulation methods cannot cope with. In particular, the chemical organization theory has been applied to biochemical domains (Centler et al., 2008b; Kaleta et al., 2006; Matsumaru et al., 2006), atmospheric photochemistries (Centler and Dittrich, 2007), and as a tool for the study of P-systems (Peter et al., 2010). It also has been proposed as a theoretical framework to design chemical computers (Matsumaru et al., 2007), and recently, for the study of social systems (Dittrich and Winter, 2008).

Thus, COT is very well suited to study autopoietic systems as both theories focus on the problem of self-maintaining organizations. At first it may seem inappropriate that a theory developed around artificial chemistries may be used to study autopoietic systems, but it should be noted that autopoietic systems are not obliged to a molecular structure or realization, that just happens to be the case of living organisms. Furthermore, the “protobio” (Varela et al., 1974) was both an early attempt to simulate autopoietic systems and an artificial chemistry. Therefore, any advance in COT might be transported directly to the theory of autopoietic systems, independent of the domain in which they are actually realized.

In this paper, we first introduce COT and its relation to autopoietic systems. Then, we present a decomposition theorem from COT and finally analyze its consequences for the long-term time behavior of biological systems.

## Autopoiesis and Chemical Organization Theory

Autopoiesis was developed as a theory for living systems by Maturana and Varela (1973). The central idea is that a living organism is a machine, constituted as a unit in space, which maintains its organization through its operation. Moreover, a living organism performs a set of pro-

cesses which generate the components necessary to realize these processes. Thus, the notion of organization as a network of interacting components which stably maintains itself in time is of most importance in this theory. Hence, a theory which concerns itself with such a concept may relate closely to autopoiesis.

The COT, which was introduced in Dittrich and Di Fenizio (2007) in the context of algebraic chemistries, is a mathematical theory, that by using the structures of sets and matrices, is able to formalize chemical reaction systems at a topological and dynamical level. In this theory, an organization is a reaction network which has the potential of being self-maintaining and thus matches very closely to the definition given by Maturana and Varela. Moreover, as “an autopoietic system is an homeostatic machine which has its organization as the variable it maintains constant”, organizations must be stable in time. The COT explores these considerations and has already had important results in this regard. In particular, in this work we present a decomposition theorem for organizations. In order to present our main result, we must first introduce the basics of COT.

## Chemical Organization Theory

### Basic Definitions

At the most basic level of this theory, we deal with two types of objects: molecular species (from now on species) and reactions. The species are the elements of a species set  $\mathcal{M} = \{m_1, \dots, m_n\}$ , and each reaction  $R$  is modeled by a pair  $R = (A, B) \in \mathcal{P}_M(\mathcal{M}) \times \mathcal{P}_M(\mathcal{M})$ , where  $\mathcal{P}_M(\mathcal{M})$  denotes the set of all the multisets formed by elements in  $\mathcal{M}$ . A multiset is defined by a pair  $(X, \eta_X)$ , where  $X$  is a set and the function  $\eta_X : X \rightarrow \mathbb{N}_0$  states the number of occurrences  $\eta_X(x)$  (multiplicity) of  $x$  in the multiset. In order to be consistent with the usual notation of chemical reactions, we will write the multiset  $(X, \eta_X)$  by  $\sum_{x \in X} \eta_X(x)x$ . Moreover, we will refer to the reaction  $R = (A, B)$  by  $R = A \rightarrow B$ , where  $A = (\mathcal{M}, \eta_A)$  and  $B = (\mathcal{M}, \eta_B)$ .

From now on, let  $\mathcal{R} = \{R_1, \dots, R_k\}$ , where  $R_i = A_i \rightarrow B_i$ , with  $A_i = a^{i1}m_1 + \dots + a^{in}m_n$  and  $B_i = b^{i1}m_1 + \dots + b^{in}m_n$ , for  $i = 1, \dots, k$  and  $j = 1, \dots, n$ .  $a^{ij}$  corresponds to the stoichiometric coefficient of  $m_j$  in reaction  $R_i$ , that is, the multiplicity  $\eta_{A_i}(m_j)$  of molecule  $m_j$  in  $A_i$ ;  $b^{ij}$  is defined in a similar way. Now we can define an Algebraic Chemistry, which captures the notion of system, as follows:

**Definition 1** An Algebraic Chemistry is a pair  $\langle \mathcal{M}, \mathcal{R} \rangle$ .

A species  $m \in \mathcal{M}$  is said to be *present* in a multiset  $(X, \eta_X) \in \mathcal{P}_M(\mathcal{M})$  if and only if its multiplicity  $\eta_X(m)$ , is greater than zero. The *reactants* and *products* of a reaction  $R = A \rightarrow B$  are the species present in  $A$  and in  $B$  respectively. The reaction  $R$  can be *fired* by a set  $X \subseteq \mathcal{M}$  if and only if all species present in  $A$  are in  $X$ .

From now on let  $X \subseteq \mathcal{M}$ . Note that there exists a maximal set of reactions  $\mathcal{R}_X \subseteq \mathcal{R}$  which can be fired by  $X$ .  $\mathcal{R}_X$

is composed by the reactions  $R_i = A_i \rightarrow B_i$  such that, if  $m$  is present in  $A_i$ , then  $m \in X$ . We call  $\mathcal{R}_X$  the *possible reactions* set of  $X$ .

In order to deal with the dynamical aspects of any system, it is desirable that the system maintains its identity. This leads to the question of whether the system, left to react for an arbitrary amount of time, will generate species which where originally absent. Note that in a general chemical setting, in which no species will be used up completely, all the reactions that can be fired will fire at some positive rate; therefore, it suffices to check if the set of possible reactions for the system produces any novel species. If it does not, we say that the set of species is closed. The following definition states this formally:

**Definition 2** We say  $X$  is closed if and only if for all  $R = A \rightarrow B \in \mathcal{R}_X$ ,  $m$  is present in  $B$  implies  $m \in X$ . Let  $G_{CL}(X)$  be the closure of  $X$ , then it is the smallest closed set containing  $X$ .

**Remark** The closure of a set has been proven to be unique (Dittrich and Di Fenizio, 2007).

Thus, any given set of species will react growing in qualitative novelty until it reaches its closure, but it is unclear whether the set will be stable in time, considering that during reactions species are consumed and their concentration could drop to zero. This motivates the study of dynamical properties of sets of species.

### Dynamical Aspects

The *stoichiometric matrix*  $\mathbf{S} = (s_{ij})$  associated with  $\langle \mathcal{M}, \mathcal{R} \rangle$  is a  $n \times k$  matrix, where  $s_{ij}$  is the stoichiometric coefficient of species  $m_i$  in the reaction  $R_j$  ( $s_{ij}$  is negative if species  $m_i$  is consumed by reaction  $R_j$ ). Indeed,  $s_{ij} = b^{ji} - a^{ji}$ . The stoichiometric matrix is at the core of current systems biology (Schuster et al., 1999; Schilling and Palsson, 1998) and its properties have been extensively studied (Kacser and Burns, 1973; Heinrich and Rapoport, 1974). Let the *flux vector*  $\mathbf{v} = (v_1, \dots, v_k)$  be a non negative vector such that the application of  $\mathbf{v}$  on the stoichiometric matrix  $\mathbf{S}$  represents a reaction process, i.e. for  $i = 1, \dots, k$ , the rate of the reaction  $R_i$  in the system is given by  $v_i$ . We define the *production rate vector* by  $\mathbf{f} = \mathbf{S}\mathbf{v}$ . Thus, for  $i = 1, \dots, n$ , we have that  $f_i$  is the rate of production of the species  $m_i$  in the reaction process determined by  $\mathbf{v}$ .

We can describe the dynamics of the species concentrations  $\mathbf{x} = (x_1, \dots, x_n)$  by the system of ODEs

$$\dot{\mathbf{x}} = \mathbf{S}\mathbf{v}(\mathbf{x}, \mathbf{k}), \quad (1)$$

where according to mass-action kinetics

$$v_i = k_i \prod_{j=1}^n x_j^{a_{ij}}$$

for  $i = 1, \dots, k$ , is the *flux*, and  $\mathbf{k} = (\mathbf{k}_1, \dots, \mathbf{k}_k)$  is a strictly positive vector denoting reaction rate constants. We call ODE (1) a *chemical reaction system*.

In order to relate the statical domain with the dynamical domain, we introduce the idea of abstractions and instances:

**Definition 3** The *abstraction* of state  $\mathbf{x}$  is the set  $\phi(\mathbf{x})$  with

$$\begin{aligned} \phi : \mathbb{R}_{\geq 0}^n &\mapsto \mathcal{P}(\mathcal{M}) \\ \mathbf{x} &\mapsto \phi(\mathbf{x}) \quad \equiv \{m_i \in \mathcal{M} : \mathbf{x}_i > \epsilon\}, \end{aligned} \quad (2)$$

where  $\mathbb{R}_{\geq 0}^n$  denotes the set of non-negative real numbers, and  $\epsilon$  is a concentration threshold. Moreover, given a set of species  $X \subseteq \mathcal{M}$ , a state  $\mathbf{x}$  is an *instance* of  $X$  if and only if its abstraction equals  $X$ .

## Chemical Organizations

The following definition is at the core of chemical organization theory:

**Definition 4** A subset of species  $X \subseteq \mathcal{M}$  is an *organization* if and only if

1.  $X$  is *closed* and
2.  $X$  is *self-maintaining*, i.e. there is a strictly positive flux vector  $\mathbf{v}$  so that

$$\mathbf{S}_X \mathbf{v} \geq \mathbf{0}$$

where  $\mathbf{S}_X$  is the stoichiometric matrix associated to the Algebraic Chemistry  $\langle X, \mathcal{R}_X \rangle$ .

Organizations are sets of species which cannot produce new species by their possible reaction set. Also, it is possible that during the operation of an organization, the concentration of none of the species decreases; thus, an organization either maintains itself in time or grows in terms of the concentration of its species. This definition shares fundamental properties with that of autopoietic systems to the extent that all autopoietic systems are organizations. Note that not all organizations are autopoietic systems as an organization that keeps growing is not homeostatic and will eventually rupture its container. This motivates the study of the fixed points and other attractors of the chemical reaction systems.

The following theorem relates fixed points and organizations<sup>1</sup>.

**Theorem 1** If  $\mathbf{x}$  is a fixed-point of the ODE (1), i.e.  $\mathbf{S}\mathbf{v}(\mathbf{x}, \mathbf{k}) = \mathbf{0}$ , then the abstraction  $\phi(\mathbf{x})$  is an organization.

Fixed points are related to the dynamic stability of chemical systems. Moreover, since fixed points determine most of the characteristics of the dynamic systems they belong to (Strogatz, 2000), Theorem 1 provides a link between the

long-term behavior of a chemical reaction system and its underlying reaction network. This allows the study of the system's dynamics by the chemical organization theory. Furthermore in (Peter and Dittrich, 2011), Theorem 1 is extended to other stable asymptotic behaviours such as periodic orbits and limit cycles. In addition, the necessary conditions for the existence of adequate flux vectors are explored in (Peter et al., 2010). Note that a fixed point in this context does not refer to thermodynamic equilibrium but to the maintenance of the size of the system in terms of the number of its components. The question about stability refers to the conservation of the structure or organization of the processes in a given timescale as the system is also subject to an evolutionary dynamic which can lead to change or desintegration. Now that we have introduced the idea of organization and shown some relevant aspects, we will focus on our main result; the decomposition theorem.

## Species Role in a Network

The idea behind the role of a species is that it can be classified in relation to a set of species by how it behaves in the set of possible reactions.

## Reactivity and Catalysts

**Definition 5** Let  $m \in X$ , then

- $m$  is *non-reactive* w.r.t  $X$  if and only if for all reactions  $R = A \rightarrow B \in \mathcal{R}_X$ ,  $m$  is not present in  $A$  nor in  $B$ .
- $m$  is a *catalyst* w.r.t  $X$  if and only if for some reaction  $R' = A' \rightarrow B' \in \mathcal{R}_X$ ,  $m$  is present in  $A$  and for all reactions  $R = A \rightarrow B \in \mathcal{R}_X$ ,  $\mathcal{A}(A, m) = \mathcal{A}(B, m)$ .
- $m$  is *reactive* w.r.t  $X$  if and only if for some reaction  $R' = A' \rightarrow B' \in \mathcal{R}_X$ ,  $m$  is present either in  $A'$  or in  $B'$  and for some reaction  $R = A \rightarrow B \in \mathcal{R}_X$ ,  $\mathcal{A}(A, m) \neq \mathcal{A}(B, m)$ .

We say that  $Y \subseteq X$  is a *non-reactive*, *catalytic* or *reactive* set of  $X$ , if for all  $m \in Y$ ,  $m$  is non-reactive, a catalyst or reactive w.r.t  $X$  respectively.

The following lemma is straightforward

**Lemma 1** There is a unique maximal non-reactive, catalytic and reactive set of  $X$ .

**Definition 6** The maximal non-reactive, catalytic and reactive sets of  $X$  are called the *non-reactive*, *catalytic* and *reactive* sets of  $X$  respectively.

## Overproduction

**Definition 7** Consider the Algebraic Chemistry  $\langle \mathcal{M}, \mathcal{R} \rangle$  and a non-negative flux vector  $\mathbf{v}$  such that  $(\mathbf{S}\mathbf{v})_i = \mathbf{f}_i \geq 0$  for  $i = 1, \dots, n$ . If  $\mathbf{f}_j > 0$  for some  $j = 1, \dots, n$ , we say that  $m_j$  is an *overproduced species* in  $\langle \mathcal{M}, \mathcal{R} \rangle$ .

<sup>1</sup>Proof can be found in (Dittrich and Di Fenizio, 2007).



Overproduced species have a positive production rate for certain flux vectors which do not lead to the consumption of any other species. We remark that on the one hand, the definition of overproduced species does not demand that the system is self-maintaining because the flux vector is only required to be non-negative, but on the other hand, overproduced species definition not only requires the non-negative production of all the species, but also the positive production of at least one species. Thus, overproduced species are the species that can be *indefinitely* produced by some *reaction pathway*. Note that although this seems to violate the law of mass conservation, real systems require a constant input of mass or energy, and thus, it is usual when simulating or analyzing chemical networks to include an outer source of mass which does not decrease when consumed by a reaction. The relevance of these species is that they can actually be overproduced without consuming any of the inner species of the system; hence, they embody the notion of input. The following lemma is straightforward.

**Lemma 2** *Let an overproduced species  $m \in X$  in  $\langle X, \mathcal{R}_X \rangle$ . If  $X \subset Y$ , then  $m$  is overproduced in  $\langle Y, \mathcal{R}_Y \rangle$ .*

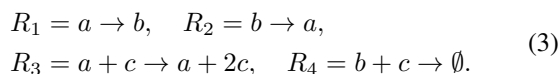
**Corollary 1** *If  $X$  is a set of overproduced species in  $\langle \mathcal{M}, \mathcal{R} \rangle$ , then its closure  $G_{CL}(X)$  is also overproduced.*

**Lemma 3** *There exists a unique maximal set  $F$  of overproduced species in  $\langle X, \mathcal{R}_X \rangle$ .*

**Proof** If there are no overproduced species in  $X$  then the maximal overproduced set is the empty set. Otherwise the set containing all the overproduced species in  $\langle X, \mathcal{R}_X \rangle$  leads to a maximal overproduced set. Now we are going to prove that the maximal overproduced set is unique. Suppose that there are two maximal overproduced sets  $F_1, F_2 \in X$  and  $F_1 \neq F_2$ , let  $\mathbf{v}_1, \mathbf{v}_2$  the flux vectors required to verify the overproduced property of  $F_1$  and  $F_2$  w.r.t  $\langle X, \mathcal{R}_X \rangle$  respectively. Trivially,  $\mathbf{v}_1 + \mathbf{v}_2$  verifies the overproduced property of  $F_1 \cup F_2$  w.r.t  $\langle X, \mathcal{R}_X \rangle$ , and  $F_i \subset F_1 \cup F_2$  for  $i = 1, 2$ . As the inclusion is strict we have a contradiction.

**Definition 8** *The maximal set of overproduced species  $F$  with respect to  $X$  is called the overproduced set of  $X$ .*

**Remark** Consider the situation of adding a species  $m$  to an organization  $O$ . The fact that  $m$  is overproduced in  $O' = O \cup \{m\}$  does not guarantee that  $O'$  is an organization. For example, consider the set of species  $O' = \{a, b, c\}$  and the set of reactions



We have that  $O = \{a, b\}$  is an organization,  $c$  is overproduced in  $O'$ , but  $O'$  is not an organization.

Then, at first sight the overproduced species could be seen as a useless definition concerning the self-maintenance of a reaction network because the overproduced species ( $c$  in the

example) can catalyze the consumption of species that cannot be recovered in the network ( $b$  through reaction  $R_4$  in the example). However, the identification of the overproduced molecules of a set  $X$  simplifies the verification of the self-maintaining condition of any set that contains  $X$ . Indeed in the example above, we have that  $c$  is overproduced in  $\{a, c\}$ , thus we can avoid the calculation of the production of the species  $c$  when verifying the self-maintenance of  $O'$ .

## Roles and Organizations

From now on let  $N, E, F$  the non-reactive, catalyst and overproduced set of  $X$  respectively.

**Definition 9**  $X - (F \cup E \cup N)$  is the potential active cycle (PAC) of  $X$  w.r.t  $F$ .

**Remark** For any given flux vector which verifies the self-maintenance of  $X$ , the PAC has a production rate equal to zero. But PAC should not be confused with the set of species with a production rate equal to zero. Indeed, the non-reactive and catalytic species have a production rate zero, but they do not belong to the PAC. The following lemma states that no species can be only produced or only consumed in the PAC of an organization:

**Lemma 4** *Let  $C$  be the PAC of  $X$ . If  $X$  is an organization, then for every  $m \in C$  we have that  $m$  is consumed by some reaction  $R \in \mathcal{R}_X$  and produced by other reaction  $R' \in \mathcal{R}_X$ .*

**Proof** Let  $m \in C$ , then  $m$  cannot be non-reactive either catalyst. As  $m$  has production zero, then  $m$  is a reactive species w.r.t  $X$ . Then, there must exist a reaction  $R = A \rightarrow B \in \mathcal{R}_X$  s.t  $\mathcal{A}(A, m) \neq \mathcal{A}(B, m)$ . If  $\mathcal{A}(A, m) > \mathcal{A}(B, m)$ , as  $X$  is an organization, there has to exist some reaction  $R' = A' \rightarrow B'$  s.t  $\mathcal{A}(A', m) < \mathcal{A}(B', m)$ . On the other hand, if  $\mathcal{A}(A, m) < \mathcal{A}(B, m)$ , as  $m$  is not overproduced (because  $m \in C$ ), there has to exist some reaction  $R' = A' \rightarrow B'$  s.t  $\mathcal{A}(A', m) > \mathcal{A}(B', m)$ .

## PAC and Dependent Connectivity

We are going to define a special notion of connectivity which will allow us to separate the PAC of a set  $X$  in a number of partially non-overlapping sub-PACs, such that the self-maintenance of  $X$  can be studied from the self-maintenance of the sub-PACs. From now on, let  $X$  be a closed set<sup>2</sup>.

The following definition of connectivity appears in (Centler et al., 2008a):

**Definition 10** *Two species  $m_i$  and  $m_j$  in  $X$  are directly connected in  $\langle X, \mathcal{R}_X \rangle$  if and only if there exist a reaction  $R = A \rightarrow B \in \mathcal{R}_X$  such that  $\{m_i, m_j\} \subseteq A \cup B$ .*

<sup>2</sup>Verifying the closed property, and obtain the closure of a set of species is trivial compared with verifying its self-maintenance (Centler et al., 2008a)

**Definition 11** Two species  $m_i$  and  $m_j$  in  $X$  are connected in  $\langle X, \mathcal{R}_X \rangle$  if and only if there exist a sequence of species  $m_0, \dots, m_p \in X$  such that  $m_i = m_0$ ,  $m_k$  and  $m_{k+1}$  are directly connected in  $\langle X, \mathcal{R}_X \rangle$  for all  $k = 0, \dots, p-1$  and  $m_p = m_j$ .

We present a more restricted notion of connectivity than definition 11. This restriction only connects species that are non-independent when verifying self-maintainance.

**Definition 12** Two species  $m_i$  and  $m_j$  in  $X$  are dependently connected in  $\langle X, \mathcal{R}_X \rangle$  if and only if there exists a sequence of species  $m_0, \dots, m_p \in X - (E \cup F)$  such that  $m_i = m_0$ ,  $m_k$  and  $m_{k+1}$  are directly connected in  $\langle X, \mathcal{R}_X \rangle$  for all  $k = 0, \dots, p-1$  and  $m_p = m_j$ .

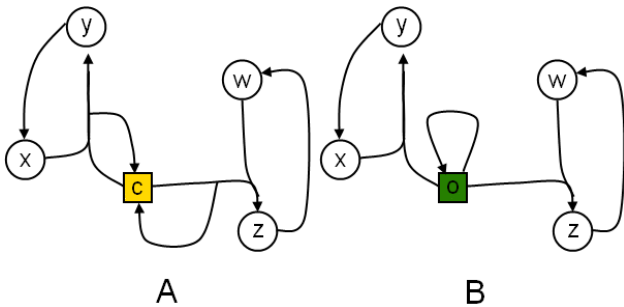


Figure 1: Following definition 11, both networks **A** and **B** are fully connected. Note that in **A**, the self-maintainance of  $C_1 = \{x, y, c\}$  and  $C_2 = \{z, w, c\}$  are independent. The same situation occurs in **B** with sets  $C'_1 = \{x, y, o\}$  and  $C'_2 = \{z, w, o\}$ . *Dependent connection* allows to connect all the species in  $C_1$  without connecting them to species in  $C_2$  and viceversa because  $C_1$  and  $C_2$  are connected through a catalyst. Analogously  $C'_1$  and  $C'_2$  are not dependently connected because they are connected through an overproduced species.

**Lemma 5** Let  $m, \bar{m} \in X$ .  $m$  is dependently connected in  $\langle X, \mathcal{R}_X \rangle$  to  $\bar{m}$  if and only if  $\bar{m}$  is dependently connected in  $\langle X, \mathcal{R}_X \rangle$  to  $m$ .

To continue, we need to mention that a computer science formalism called Petri Nets (Murata, 1989), has been considered as an interesting source of insights for the biochemical pathways research (Reddy et al., 1993, 1996). Petri Nets arose from the necessity to formalize concurrent processes. We will incorporate some fundamental topological parameters of Petri Nets to our analysis: the set of input transitions of a place and the set of input places of a transition.

**Definition 13** Let  $m \in \mathcal{M}$ . We define

$$\text{Act}(m, \mathcal{R}) = \{R = A \rightarrow B \in \mathcal{R} \mid m \text{ is present in } A\}.$$

We say  $\text{Act}(m, \mathcal{R})$  is the activable set of reactions of  $m$  in  $\mathcal{R}$ .

**Definition 14** Let  $R = A \rightarrow B \in \mathcal{R}$ . We define

$$\text{Req}(R) = \{m \mid m \text{ is present in } A\}.$$

We say  $\text{Req}(R)$  is the required set of species of  $R$ . Furthermore, for a set of reactions  $S \subseteq \mathcal{R}$  we define

$$\text{Req}(S) = \bigcup_{R \in S} \text{Req}(R).$$

The set of input places in Petri Nets corresponds to the set  $\text{Req}(\cdot)$ , and the set of input transitions corresponds to  $\text{Act}(\cdot, \mathcal{R})$ .

**Definition 15** We define  $\text{Causal}^*(m, \mathcal{R}_X)$  as the set of dependently connected species in  $\langle X, \mathcal{R}_X \rangle$  to  $m$ , and  $\text{Causal}(m, \mathcal{R}_X) = \text{Req}(\text{Act}(\text{Causal}^*(m, \mathcal{R}_X)))$ .

The following lemmas are derived straightforward from lemma 5 and definition 15:

**Lemma 6** Let  $m, \bar{m} \in \mathcal{M}$ .  $\bar{m} \in \text{Causal}^*(m, \mathcal{R}_X)$  if and only if

$$\text{Causal}^*(m, \mathcal{R}_X) = \text{Causal}^*(\bar{m}, \mathcal{R}_X).$$

**Lemma 7** Let  $R \in \mathcal{R}_X$ ,  $m, \bar{m} \in X - (E \cup F)$  s.t.  $m \notin \text{Causal}(\bar{m}, \mathcal{R}_X)$ . If  $R \in \text{Act}(\text{Causal}(m, \mathcal{R}_X) \cap \text{Causal}(\bar{m}, \mathcal{R}_X), \mathcal{R}_X)$  then  $R \in \mathcal{R}_{E \cup F}$ .

$\text{Causal}^*(\cdot, \cdot)$  provides a way to split a set  $X$  of species in dependent connected subsets. It is necessary to identify a the catalytic set  $E$  and the overproduced set  $F$  to generate such separation. The more elements are in  $E \cup F$ , the more chance of recognize the independent causal connected sets we have.

**Lemma 8** Let  $D$  be the PAC of  $X$ . Then

$$D = \bigcup_{m \in D} \text{Causal}^*(m, \mathcal{R}_X)$$

**Proof** Note that  $D \subseteq \bigcup_{m \in D} \text{Causal}^*(m, \mathcal{R}_X)$ . Let  $m \in$

$\bigcup_{m' \in D} \text{Causal}^*(m', \mathcal{R}_X)$ , then for some species  $m' \in D$  we have  $m$  is dependently connected to  $m'$ , then  $m'$  is also dependently connected to  $m$ . This means  $m$  is a reactive, non overproduced, and non-catalytic species. Then  $m \in D$ .

From now on we let  $D \subseteq X$  be the potential active cycle of  $X$ .

**Definition 16** Any set  $D' \subseteq D$  s.t.  $D = \bigcup_{m \in D'} \text{Causal}^*(m, \mathcal{R}_X)$  is called a base of  $D$ . Any minimal cardinality base of  $D$  is called a minimal base of  $D$ .

**Lemma 9** Let  $D', D''$  be two minimal bases of  $D$ . Then every species in  $D'$  is dependently connected to one and only one species of  $D''$ .

**Proof** Let  $m \in D'$  and suppose that there is no species in  $D''$  dependently connected to  $m$ . By corollary 6 we have  $Causal^*(m, \mathcal{R}_X)$  is not contained in  $\bigcup_{m' \in D''} Causal^*(m, \mathcal{R}_X)$ . Then there has to be at least one species dependently connected to  $m$  in  $D''$ . Now suppose there is more than one species dependently connected to  $m$  in  $D''$ . Let  $m_1, m_2 \in D''$  such species. As  $m_1$  and  $m_2$  are dependently connected to  $m$ , then  $m_1$  and  $m_2$  are dependently connected. By corollary 6 we have  $Causal^*(m_1, \mathcal{R}_X) = Causal^*(m_2, \mathcal{R}_X)$ . Then  $D''$  is not a minimal base of  $D$ .

A minimal base of  $D$  is a set which generates all the non dependent sub-PACs of the  $D$ . We are going to prove that the self-maintainance of a potential active cycle can be obtained from the self-maintainance of its non dependent sub-PACs.

**Lemma 10** Let  $D'$  a minimal base of  $D$ . Then

$$Act(D, \mathcal{R}_X) = \bigcup_{m \in D'} Act(Causal^*(m, \mathcal{R}_X), \mathcal{R}_X).$$

**Proof** Note that

$$\bigcup_{m \in D'} Act(Causal^*(m, \mathcal{R}_X), \mathcal{R}_X) \subseteq Act(D, \mathcal{R}_X).$$

Let  $R \in Act(D, \mathcal{R}_X)$  then for some  $m \in D$  we have  $R \in (m, \mathcal{R}_X)$ . From definition 16 we have that there is  $m' \in D'$  s.t  $m \in Causal^*(m', \mathcal{R}_X)$ . Then by corollary 6 we have

$$R \in Act(Causal^*(m', \mathcal{R}_X), \mathcal{R}_X) \subseteq \bigcup_{m \in D'} Act(Causal^*(m, \mathcal{R}_X), \mathcal{R}_X).$$

## Decomposition Theorem for Organizations

**Theorem 2** Let  $D' = \{\bar{m}_1, \dots, \bar{m}_d\}$  a minimal base of  $D$ . For  $i = 1, \dots, d$  let

$$D_i = Causal(\bar{m}_i, \mathcal{R}_X), \\ F_i = Causal(\bar{m}_i, \mathcal{R}_X) \cap F.$$

Let  $\emptyset \rightarrow Y = \{\emptyset \rightarrow y / y \in Y\}$ .  $X$  is self-maintaining if and only if for all  $i = 1, \dots, d$  we have that  $D_i$  is self-maintaining in the subnetwork  $\langle D_i, \mathcal{R}_{D_i} \cup \emptyset \rightarrow F_i \rangle$ .

**Proof**  $\Rightarrow$ : Let  $\bar{F} = \bigcup_{i=1}^d F_i$ . Note that  $X$  is self-maintaining in  $\langle X, \mathcal{R}_X \rangle$  if and only if  $X$  is self-maintaining in  $\langle X, \mathcal{R}_X \cup \emptyset \rightarrow \bar{F} \rangle$ . Let  $\mathbf{v}$  be a vector which verifies the self-maintainance of  $X$  in  $\langle X, \mathcal{R}_X \rangle$ . Let  $Act(D_i, \mathcal{R}_X) = \{R_{\alpha_1}, \dots, R_{\alpha_i}\}$ , then  $\bar{\mathbf{v}}$  lead to a non-negative production on all the species of  $Causal^*(\bar{m}_i, \mathcal{R}_X)$  where

$$\bar{\mathbf{v}}_i = \begin{cases} \mathbf{v}_i & \text{if } i = \alpha_j \text{ for some } j, \\ 0 & \text{else} \end{cases}$$

As the rest of species belong to  $\bar{F}$ , to reach their non-negative production we use the reactions in  $\emptyset \rightarrow \bar{F}$ .

$\Leftarrow$ : Let  $\mathbf{v}_1, \dots, \mathbf{v}_d$  the flux vectors which verifies the self-maintainance of  $\langle D_i, \mathcal{R}_{D_i} \cup \emptyset \rightarrow F_i \rangle$ ,  $i = 1, \dots, d$  and  $\mathbf{v}^F$  the flux vector which verifies the potential overproduction of  $F$  w.r.t  $X$ . Then there exist a non-negative number  $\beta$  s.t  $\beta \mathbf{v}^F + \sum_{i=1}^d \bar{\mathbf{v}}_i$  verifies the self-maintainance of  $X$ , where  $\bar{\mathbf{v}}_i$  is the flux  $\mathbf{v}_i$  represented as a flux vector for  $\mathcal{R}_X$ , i.e. completed with zeros in the coordinates representing reactions that are not in  $\mathcal{R}_{D_i}$ .

**Corollary 2** Let  $D'$  a minimal base of  $D$ . Then  $X$  can be non-overlapping decomposed (partitioned) as

$$X = N \cup E \cup F \cup D_1^* \cup \dots \cup D_d^*. \quad (4)$$

With  $D_i^* = Causal^*(m_i, \mathcal{R}_X)$ , and  $m_i$  the  $i$ -th element of  $D'$ . Moreover  $X$  is self-maintaining if and only if  $E \cup F \cup D_i^*$  is self-maintaining for  $i = 1, \dots, d$ .

## Stability, COT and Autopoiesis

Living organisms are systems far from thermodynamic equilibrium, therefore the question about stability refers to the conservation of the organization of the organism's processes. This issue requires some attention, as the processes an autopoietic system are homeostatic in essence, but they are always potentially subject to dramatic changes. A clear example of this is the cell cycle, which is driven by cyclic, i.e. non stationary, processes like the cyclins proteins family expression patterns. In relation to this, the autopoiesis theory describes living organisms as processes that produce the components that give rise to those processes, where some attributes are preserved and others may change (Varela et al., 1974) allowing a structural drift (Maturana and Mpodozis, 2000). Therefore, in a way, autopoietic systems are not obliged to exhibit stability in the long run.

The basis of structural drift in biochemical networks is metabolic regulation. This is managed by the modulation of enzymes expression and also by means of a co-catalysis phenomenon in which coenzymes and regulators interact with the enzyme structure-function relation. This metabolic regulation determines the cell's developmental *direction* between a wide range of possible organizations. In order to illustrate this idea, imagine a system *following* an attractor when suddenly the concentration of a given regulator reaches a level that triggers a deep change in the structure of the network. Now, the attractor mentioned above is absent and the system takes a different pathway in which another phase space shift can occur. If this process becomes cyclic, the entire loop can be described as a limit cycle, but a decomposition of phase space in a set of *contextual* mini phase spaces that emerge in different regulatory scenarios could contribute both to the understanding of biological operations as to the algorithms research in biology inspired AC simulations. Therefore, in the COT, the changes in structure can be represented as changes in the phase space, leading to a dependence of phase space with the state of enzymes and regulators.

Some future goals in biologically inspired AC are the study of cyclic behaviour and autopoiesis theory's structural coupling. Preliminary studies concerning artificial autopoietic systems have been done in (Peter et al., 2010), by making use of the P-systems formalism. A P-system is formed by a set reaction networks, each one enclosed by a membrane. The reaction networks can interact diffusing particles through membranes. In particular, it is shown how a bistable cyclic process can be reached among two different unstable reaction networks, by exchanging in low rates their subproducts, i.e. forming together an organization.

## Conclusions

In this paper we have shown how the chemical organization theory connects deeply to notions of autopoietic systems. Moreover, every autopoietic system is an organization, and thus, theorems derived for organizations are valid for autopoietic systems.

We introduce the notion of the role played by a species in a subnetwork of a reaction network. The different possible roles that a species can play in a subnetwork (non-reactive, catalyst, overproduced, active cyclic) give information about the structure of the subnetwork (lemmas 1, 3 and 4). We also introduce the notion of dependent connectivity in a reaction network (definition 12), which is useful to split a reaction network into the minimal parts required to verify the self-maintainance (Theorem 2). This theorem helps to simplify the organization verification not only by decomposing the set, but also when we keep track of the decomposition of the set to verify the self-maintainance of sets that contain it.

The fact that an organization can be subdivided into self-maintaining subnetworks which are mostly independent one from another (and not necessarily closed), is both striking and noteworthy. The decomposition theorem stated in this work shows that the long-term behaviour of an organization can depend on sets of species whose states are weakly coupled. This result opens new paths of analysis for a broad set of fields, from metabolic dynamics to ecological networks. This result also relates directly to a debated subject which is the composition of autopoietic systems by other autopoietic systems.

At this point a comment on the domain of applicability of this theorem is convenient. On some cases, like reactive flow systems where there is a spontaneous decay of every species, the decomposition is trivial, i.e., the system cannot be further subdivided. This is because each species of an organization would be overproducible (to counteract the decay). So, this paper addresses those systems where some species take part in reactions but do not decay spontaneously. In the case of living systems, every molecule decays spontaneously or, equivalently, dilutes when the system grows. This seems to make an argument towards the unapplicability of the decomposition theorem to living systems. However, the fact that every molecule decays in a

living system is only relevant at the right timescale. For a smaller timescale some molecules do not decay. Therefore, it is important to examine living systems at different timescales. Choosing a very long time scale, basically no system would continue to exist so that autopoiesis would not become visible. While with a smaller timescale more and more elements (molecules) would become stable and would not decay spontaneously. Here is where the decomposition can potentially be applied.

This decomposition theorem is suggested as a starting point for the complexity analysis of the organization verification problem as well as for the classification of reaction networks in terms of how complicated it is to compute their organizational structure. Although this is a relevant result for current systems biology as it simplifies analysis and simulations, and for artificial life, at it may guide the construction of artificial self-sustaining systems, it also demands revisiting questions on biological systems concerning their modularity, autonomy, and even the notion of their identity.

## References

- Benkő, G., Centler, F., Dittrich, P., Stadler, B. M. R., and Stadler, P. F. (2009). A topological approach to chemical organizations. *Artif. Life*, 15:71–88.
- Centler, F. and Dittrich, P. (2007). Chemical organizations in atmospheric photochemistries: A new method to analyze chemical reaction networks. *Planet. Space Sci.*, 55:413–428.
- Centler, F., Kaleta, C., Di Fenizio, P. S., and Dittrich, P. (2008a). Chemical organizations in a toy model of the political system. *Advances in Complex Systems*, 11(4):609–627.
- Centler, F., Kaleta, C., Di Fenizio, P. S., and Dittrich, P. (2008b). Computing chemical organizations in biological networks. *Bioinformatics*, 24:1611–1618.
- Cornish-Bowden, A., Cárdenas, M. L., Letelier, J. C., and Soto-Andrade, J. (2007). Beyond reductionism: Metabolic circularity as a guiding vision for a real biology of systems. *PROTEOMICS*, 7:839–845.
- Dittrich, P. and Di Fenizio, P. S. (2007). Chemical organization theory. *Bulletin of Mathematical Biology*, 69:1199–1231.
- Dittrich, P. and Winter, L. (2008). Computing chemical organizations in biological networks. *Bioinformatics*, 24(14):1611–1618.
- Eigen, M. and Schuster, P. (1977). A principle of natural self-organization. *Naturwissenschaften*, 64:541–565.
- Fontana, W. and Buss, L. (1994). The arrival of the fittest: Toward a theory of biological organization. *Bulletin of Mathematical Biology*, 56:1–64.
- Heinrich, R. and Rapoport, T. A. (1974). A linear steady-state treatment of enzymatic chains. *European Journal of Biochemistry*, 42:89–95.
- Hordijk, W. and Steel, M. (2004). Detecting autocatalytic, self-sustaining sets in chemical reaction systems. *Journal of Theoretical Biology*, 227:451–461.

- Jaramillo, S., Honorato-Zimmer, R., Pereira, U., Contreras, D., Reynaert, B., Hernández, V., Soto-Andrade, J., Crdenas, M., Cornish-Bowden, A., and Letelier, J. C. (2010). (m,r) systems and raf sets: Common ideas, tools and projections. In Fellermann, H., Drr, M., Hanczyc, M. M., Laursen, L. L., Maurer, S., Merkle, D., Monnard, P. A., Stoy, K., and Rasmussen, S., editors, *Artificial Life XII*, pages 94–100. MIT Press, Cambridge, MA.
- Kacser, H. and Burns, J. A. (1973). The control of flux. *Symposia of the Society for Experimental Biology*, 27:65–104.
- Kaleta, C., Centler, F., and Dittrich, P. (2006). Analyzing molecular reaction networks from pathways to chemical organizations. *Molecular Biotechnology*, 34:117–123.
- Kauffman, S. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22:437–467.
- Letelier, J. C., Marin, G., and Mpodozis, J. (2003). Autopoietic and (M,R) systems. *Journal of Theoretical Biology*, 222:261 – 272.
- Matsumaru, N., Centler, F., di Fenizio, P., and Dittrich, P. (2006). Chemical organization theory applied to virus dynamics. *Information Technology*, 48:154–160.
- Matsumaru, N., Centler, F., di Fenizio, P., and Dittrich, P. (2007). Chemical organization theory as a theoretical base for chemical computing. *International Journal on Unconventional Computing*, 3:285–309.
- Maturana, H. and Mpodozis, J. (2000). The origin of species by means of natural drift. *Revista chilena de historia natural*, 73(2).
- Maturana, H. and Varela, F. (1973). *De Máquinas y Seres Vivos: Una teoría sobre la organización biológica*. Editorial Universitaria.
- Murata, T. (1989). Petri nets and its applications. *Proceedings of the IEEE*, 77:541–580.
- Peter, S. and Dittrich, P. (2011). On the relation between organizations and limit sets in chemical reaction systems. *Advances in Complex Systems*, 14:77–96.
- Peter, S., Veloz, T., and Dittrich, P. (2010). Feasibility of organizations - a refinement of chemical organization theory. In: *M. Gheorghe, T. Hinze, G. Paun (Eds.), Proc. of the Eleventh International Conference on Membrane Computing*, pages 369–382.
- Reddy, V. N., Liebman, M. N., and Mavrovouniotis, M. L. (1996). Qualitative analysis of biochemical reaction systems. *Comput Biol. Med.*, 26:9–24.
- Reddy, V. N., Mavrovouniotis, M. L., and Liebman, M. N. (1993). Petri net representations in metabolic pathways. *ISMB-93 Proceedings*, pages 328–336.
- Rosen, R. (1958). A relational theory of biological systems. *Bulletin of Mathematical Biophysics*, 20:317–341.
- Schilling, C. H. and Palsson, B. O. (1998). The underlying pathway structure of biochemical reaction networks. *Proceedings of the National Academy of Sciences*, 95:4193–4198.
- Schuster, S., Dandekar, T., and Fell, D. A. (1999). Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. *Trends Biotechnology*, 17:53–60.
- Strogatz, S. (2000). *Nonlinear Dynamics and Chaos*. Westview Press, Cambridge.
- Varela, F., Maturana, H., and Uribe, R. (1974). Autopoiesis: The organization of living systems, its characterization and a model. *Biosystems*, 5:187–196.
- von Bertalanffy, L. (1968). *General System Theory: Foundations, Development, Applications*. George Braziller, New York.
- Wiener, N. (1948). *Cybernetics: Or Control and Communication in the Animal and the Machine*. Hermann and Cie Editeurs, Paris.

# The Elongation Catastrophe in Physical Self-Replicators

Nathaniel Virgo<sup>1</sup>, Chrisantha Fernando<sup>1,2</sup>, Bill Bigge and Phil Husbands<sup>1</sup>

<sup>1</sup>Department of Informatics, University of Sussex, Falmer, Brighton BN1 9RH, UK

<sup>2</sup>MRC National Institute for Medical Research, London NW7 1AA, UK

ctf20@sussex.ac.uk

## Abstract

An insufficiently appreciated paradox in the origin of life is that the replication of information-carrying molecules requires the molecules to be very specifically shaped; but such specific molecules are hard to produce without natural selection. We demonstrate and investigate this problem by building a physical model of self-replication out of specifically shaped plastic pieces with embedded magnets, which float around on an air-hockey type table. We use a mechanism known as template replication, which works by the joining of complimentary strands, roughly analogous to the biological replication of DNA, except without the involvement of enzymes. Building a physical rather than a computational model forces us to confront several issues that have analogues in the microscopic, chemical world. In particular, in order to achieve a low mutation rate we must reduce as much as possible the formation of incorrect sequences, which can happen both spontaneously and as a result of strands joining in a misaligned way. The latter results in ever-lengthening sequences in a process known as the “elongation catastrophe”. We present an overview of our design process, illustrating the many interdependent adaptations that had to be made to the pucks’ shapes in order to solve these problems while maintaining a high rate of template replication. The chicken and egg question is how, in the pre-biotic world, could template replication be achieved without the presence of enzymes that require template replication in the first place? By building a real physical model a new answer to this question is suggested. We propose that early pre-biotic monomers required structural specializations that reduced the rate of formation of incorrect sequences, without the need of an encoded enzyme.

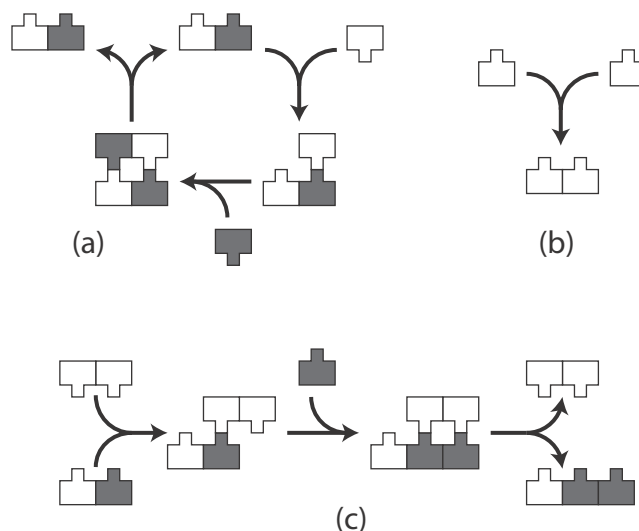
## Introduction

In the highly evolved biology of today a complex array of encoded enzymes is necessary for the replication of DNA and RNA polymers. These enzymes were not available at the origin of life, and so nucleotide template replication had to be non-enzymatic (Szathmáry, 2000; 2006). The best example of non-enzymatic template replication we have so far is still the work of Guenter von Kiedrowski (1986) who made the first non-enzymatic template replicator consisting of the hexanucleotide sequence GGCGCC that catalyses the templated ligation of CGG and CCG trimers.

In such experiments, replication must be carefully distinguished from spontaneous self-assembly which is typically easier to achieve than replication in stochastic systems. In the von Kiedrowski experiment there is a low rate of self-assembly (specifically elongation/dimerisation) by non-templated ligation of CCG and CGG. To prove replication one must compare the rate of formation of

GGCGCC in the absence and the presence of an initial seed of GGCGCC. The difference is the extent of true templated self-replication. Whereas self-assembly of random novel oligomers is fine for random search in sequence space, self-replication is crucial for evolution by natural selection, i.e. the production of offspring whose fitness correlates with parental fitness (Price, 1970). If most of the DNA in a proto-organism was self-assembled de-novo into random sequences and not replicated from the parent, the genome would be real garbage, as opposed to inherited junk.

This raises a paradox that is of no lesser importance than Eigen’s paradox regarding the error catastrophe (Eigen 1971). Our logically anterior paradox deals with the fact that specificity of self-replication over self-assembly is a critical pre-requisite for an evolvable physical template self-replicating system. Without specific ligation, random de novo synthesized sequences invade a population of replicating evolved sequences. These random sequences compete with evolved sequences for monomer resources thus diluting out evolved information (i.e. sequences that had arisen from a



**Figure 1:** A generic illustration of template replication and two side reactions that must be avoided. (a) Homologous template directed ligation (self-replication) results in the correct duplication of a sequence. (b) A new (incorrect) sequence is formed by non-templated spontaneous ligation. (c) Elongation of the original sequence by partially homologous template ligation at staggered ends. See (Fernando, Von Kiedrowski et al. 2007) for a full analysis of the elongation catastrophe.

lineage of template replication events). In addition evolved sequences become trapped inside elongating strands (that cannot easily unzip or denature) such that they cannot easily experience another round of replication, see Figure 1. We call this the elongation catastrophe and it raises what we will call the elongation paradox (Fernando, von Kiedrowski et al. 2007). How can specific ligation be achieved without complex enzymes that require template replication with specific ligation in the first place?

The minimal unit of template replication is a dimer (i.e. a polymer of length two) that can replicate three possible sequences, AB, BA, or AA(BB), as in Figure 1. The minimal unit of template replication has the capacity to replicate the specific configuration that it is in. It is this fact that allows template replication to potentially convey an unlimited amount of information (Szathmáry and Maynard Smith 1997) because of the compositionality of the genome (Fodor and Pylyshyn 1988) and to be evolvable due to the capacity for micro-mutation, i.e. small changes in the composition can generate correlated fitness variants (Price 1970). But there is a real danger with such a system that if ligation is not tightly controlled then novel random sequences can arise and evolved sequences can elongate (but not replicate) without limit, as in Figure 1(c).

Mutations must be able to occur in an evolvable system, but they must occur at a low rate in order to avoid Eigen's (1971) error threshold. A minimal evolvable system must therefore exhibit the replication of dimers with low rate of assembly of incorrect or elongated sequences. For this project we set ourselves the goal of producing a system where the average rate of replication of a seed dimer is greater than the rate of formation of all other sequences put together.

Interestingly, this elongation catastrophe was the fate of a 2D macroscopic system designed by Jarle Breivik for template replication that was faithful to some aspects of chemistry such as stochasticity and binding properties (Breivik 2001). He used 2D plastic shapes with embedded magnets and an oscillating temperature water bath. Unfortunately, despite the obvious ingenuity of the design, the original templates formed in an unseeded manner by spontaneous aggregation of "hydrogen bonded" pairs to form a double strand and no kinetic comparison between self-assembly and self-replication was made. From Figure 3 in Breivik's paper it appears that free ligation was responsible for the production of all the oligomers in that model by de novo synthesis of monomers in weakly bonded pairs. Strangely, the h-bonded pairs catalyze double p-bond formation, see Figure 3 in (Breivik 2001). It seems, no template replication was demonstrated, and if it did exist, it seems to occur much more slowly than the spontaneous formation of novel sequences. This is a problem for evolution by natural selection, not a feature. Breivik's system suffers severely from the elongation catastrophe and therefore could not be extended to undergo natural selection of sequences.

In fact, until now, to our knowledge it is still only the geneticist Lionel Penrose and his son Roger Penrose (Penrose and Penrose 1957) who have shown a relatively specific type of ligation reaction in a physical system without resorting to electronic switches and other features that make specificity of ligation trivial and thus reduce their utility in abduction to chemistry or the potential for later miniaturization (Groß,

Küchler et al. 2009). Penrose's devices use only gravity, collision, friction, and (passive and active) hooking. In the simplest model, two kinds of solid object A and B are agitated horizontally on a straight track. If seeded with either a AB or a BA dimer (AA and BB dimers cannot form in the Penroses' system) other monomers join together by being appropriately tilted, to form the identical dimer type, without novel AB or BA forms appearing spontaneously by un-catalysed ligation. E.F. Moore wrote of Penrose's design "If the reader attempts the problem of how to design the shapes of the units A and B so as to have the specified properties, the difficulties he will encounter in his attempt will cause him to more readily appreciate the ingenuity of Penrose's very simple solution to this problem." (Moore 1962).

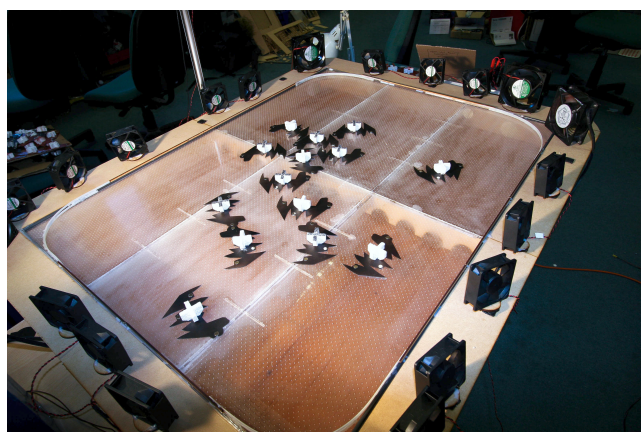
However, 1D systems are severely limited in terms of extendibility to longer sequences to achieve unlimited heredity (Szathmáry and Maynard Smith 1997) because i. they may be constrained by the initial sequence of monomers along the chain (which is a problem if the identity of monomers cannot flip between A and B, which in some of Penrose's designs they can), and ii. information about the identity of units on the inside of a sequence must pass through all other bordering units before they can influence external monomers. Again, Lionel Penrose already carefully considered information transmission through units agitated in 1D, for example he invented in a length-dependent end-blocking device that prevents anything larger than 4-mers from forming, so avoiding the elongation catastrophe in one dimension. A more complete 1D self-replicator (still 1D because it is only agitated in the horizontal axis) was later invented by Penrose to allow the replication of dimers with more possible states/configurations defined by the arrangement of hooks stacked in the 2D axis orthogonal to the axis of agitation rather than perpendicular to that axis (Penrose 1959). So, in short, Penrose took the elongation catastrophe rather seriously.

Here for the first time we present a mechanical 2D stochastic self-replicator that has limited rates of non-catalysed spontaneous self-assembly (ligation) of monomers, and limited partial homologous templated ligation. Reducing the rates of these two side-reactions serves to some extent to curtail the elongation catastrophe. However, we note that our solution is hand-designed and partial. The elongation paradox is still not solved for the origin of life, i.e. we do not know how such infra-biological monomers could have arisen with these very specific capabilities; speculation on this based on this work is given in the conclusions.

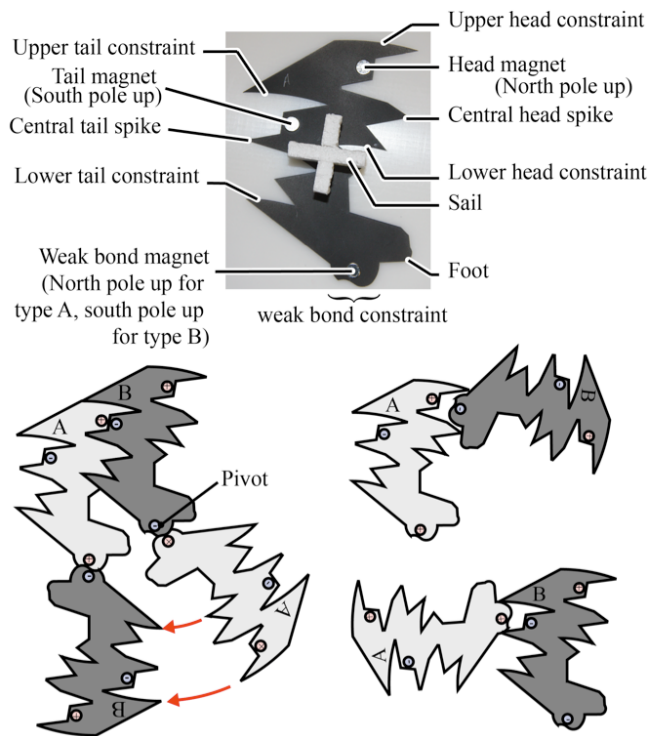
We built plastic monomers containing magnets and passive hooks and sails, that floated on an air-hockey table, and were blown by fans on the perimeter of the table, see Figure 2. Spontaneous elongation (untemplated ligation) was reduced by careful design of the physical equivalent of the phosphodiester bond. In addition, partial homologous ligation was reduced by careful design of the template complex.

Indeed, our system is a macroscopic close relative of von Kiedrowski's hexanucleotide replicators, because we have faced similar design challenges as in real chemistry, such as cyclisation and product inhibition. Guenter von Kiedrowski had to block the ends of his hexamers to prevent partial homologous ligation from catastrophically extending strands and depleting matter from the replicator cycle (Von

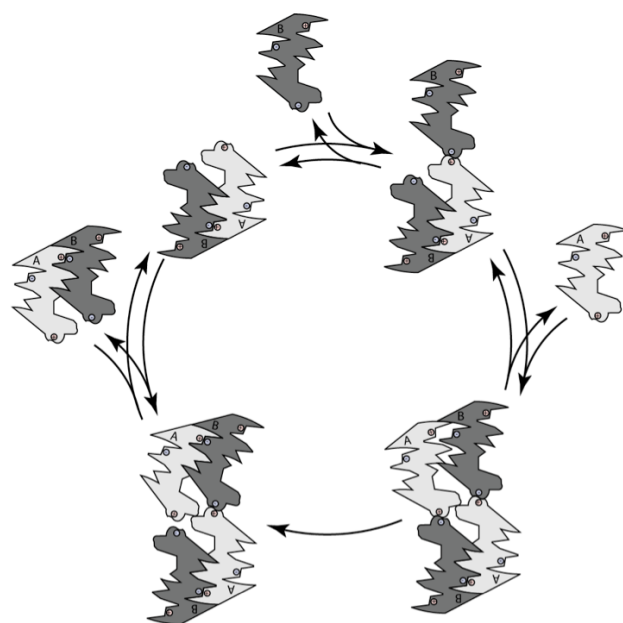




**Figure 2:** The design of the air-hockey style table containing the monomers. Sails on each monomer are blown by a perimeter of small fans. Another fan below the table passes air through small holes to suspend monomers over the table like small hovercraft.



**Figure 3:** The design of the monomers. The top photograph shows the names used in the text for important parts of the design. In photographs the two monomers are distinguished by the colour of their polystyrene sails (white for A, black for B), whereas in diagrams, type B is shown in a darker shade of grey. The lower-left diagram shows the mechanism by which templated ligation takes place (but see also Figure 4). The lower right diagram shows how the design prevents the weak bond magnets from bonding to the strong bond magnets.

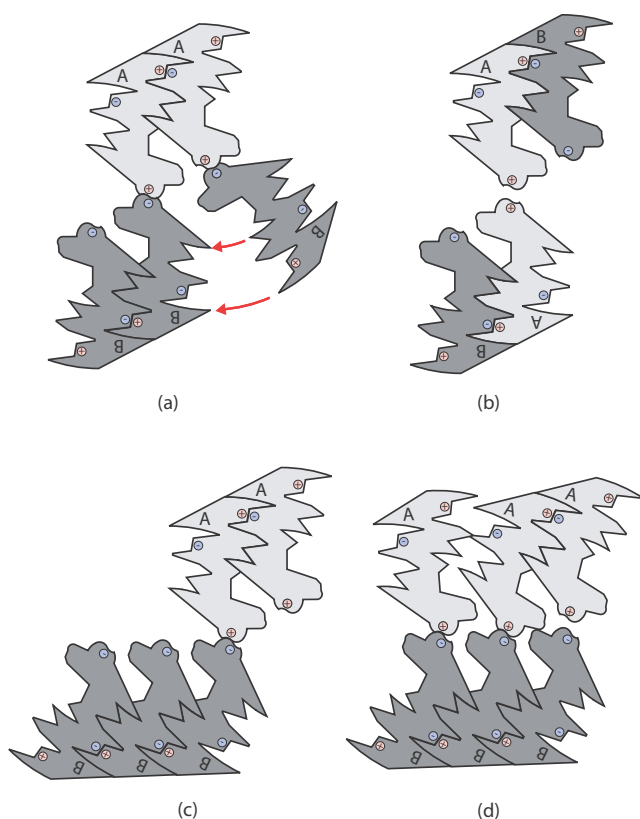


**Figure 4:** The autocatalytic cycle for replication of an AB dimer. (a) A type 'B' monomer joins to the dimer. (b) A type 'A' monomer joins to the other h-bond and swivels into place via the mechanism shown in Figure 1. Catalysis can also take place if the monomers join in the opposite order; in this case both monomers must swivel on their weak bonds, which often occurs when the configuration collides with another object. (c) A p-bond is formed by template directed ligation and, simultaneously, one of the h-bonds is broken. A collision with another molecule or the table edge is required in order for this step to occur. (d) Another collision breaks the remaining weak bond, and the two strands separate, completing the cycle.

Kiedrowski 1986). However, in our system we have not explicitly blocked the ends, but have designed all the monomers so that end-blocking is 'emergent'.

The primary advantage of a physical system over a computer simulation is it forces us to confront the problems of template replication by changing the design of the monomers, rather than by changing the simulation to reduce the problems. Similarly, while real chemical monomers can have mechanically-implemented internal states, our self-imposed restriction of no electronic components prevents us from being able to implement any arbitrary mechanism, regardless of how easily it could be implemented mechanically in chemical systems.

Next we describe the design of the pucks (monomers) and then we conduct a classical seeding experiment to distinguish self-replication from self-assembly. This is the first demonstration of a 2D template replication system that is capable of low rates of spontaneous elongation yet high rates of self-replication (without the use of monomers containing electronically implemented finite state machines).



**Figure 5:** (a) The formation of a BBB trimer due to partial homologous ligation. The production of AAA and BBB trimers in this way is relatively common in our system (see Figure 9). (b) Staggered bonding is not possible between two AB dimers (or two BA dimers) because it would require the formation of an h-bond between magnets of the same polarity. (c) It is in theory possible for a further partial homologous ligation to extend a BBB trimer into a BBBB 4-mer. However, we did not observe this in any of our trials. We suspect this is because the two polymers have a high moment of inertia about the weak bond's pivot point, destabilising the bond and making it likely to break. (d) Polymers of length greater than two cannot replicate in the same way as dimers, because the "foot" mechanism does not allow the strong bond constraints to align with the weak bond pivot.

## Methods

A frictionless table, similar to those used for air-hockey, was purpose built and consisted of a flat plastic surface perforated with an array of 1.5mm diameter holes, spaced at intervals of 10mm. An enclosure underneath this surface was pressurized with a powerful fan to produce a steady jet of air from each hole, allowing suitably shaped objects to float above the table surface. Surrounding the table was a set of approximately 20 small fans that could be arranged to cause a stochastic motion of the pucks, albeit with a significant rotational element, see Video A in Supplementary Material. There was no "temperature" oscillation as in Breivik's experiment, i.e. the fans always rotated at the same speed. The walls of the table

allowed approximately elastic collisions. The puck design is shown in Figure 3.

Pucks are 1.5mm thick and made of plastic. The bases of the pucks are flat allowing a hovercraft type low friction floating of the puck above the table. The pucks were fabricated using a Versalaser cutter. Rapid fabrication of new designs was possible for prototyping. Pucks contain molybdenum disc magnets that can be oriented with the north or south pole facing upwards, allowing specification of attractive or repulsive interaction pairs.

The final design has the following features. The strong 'phosphodiester bonds' must not form spontaneously. This is achieved by embedding the magnets deep within the puck and producing a lock and key type join which can only form if the pucks collide at a very specific orientation. This orientation tends to occur only when the two monomers are 'hydrogen bonded' to a dimer template, and not when two pucks collide against each other as untemplated monomers. Once the pucks make the p-bond the magnets are very close together so the bond is strong. Thus the p-bond is difficult to form due to steric constraints but once formed is strong due to close magnets and mechanical rigidity. The h-bonds consist of an interaction between magnets that are further apart when the bond is formed, i.e. the bond is weaker. Also, there is a curve on the surface of the bond to allow pucks to rotate when h-bonded. This rotation brings the two h-bonded monomers into the appropriate configuration for the p-bond to form.

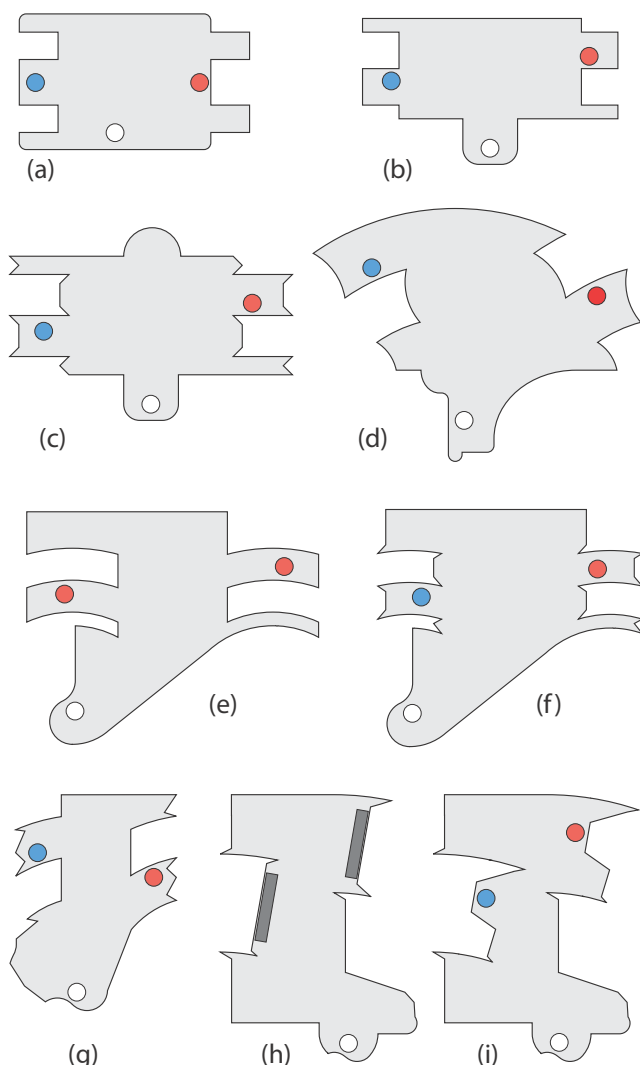
To reduce product inhibition, the pucks are shaped in such a way that two p-bonded dimers cannot be joined at both h-bonds. Thus, as the p-bond forms it breaks one of the two hydrogen bonds. The remaining h-bond is sufficiently weak that the two dimers can separate and undergo another round of replication.

There are two types of monomer, labelled 'A' and 'B', which differ only in the orientation of the magnets that form their h-bonds. 'A' type monomers can only form weak (h-)bonds with 'B' type monomers, and vice versa. Strong (p-)bonds can be formed between any pair of monomers, giving rise to four types of strong-bonded dimer, 'AA', 'AB', 'BA' and 'BB'. Template replication produces a new dimer that is both the complement and the reverse of the original. This results in three separate autocatalytic cycles: {AB}, with the reaction  $AB + A + B \rightarrow 2AB$ ; {BA}, with the reaction  $BA + A + B \rightarrow 2BA$ ; and {AA, BB} with the reactions  $AA + 2B \rightarrow AA + BB$  and  $BB + 2A \rightarrow AA + BB$ .

Misalignment with the generation of a staggered or dangling end as they are often called, can cause 'AA' dimers to be extended via catalysis to 'AAA' dimers, and similarly for the 'BB' type, by partial homologous ligation (see Figure 5). However, in all the experiments conducted we did not observe the production of 4-mers by partial homologous ligation. Importantly misalignment did not tend to occur for 'AB' and 'BA' dimers, which cannot catalyse partial homologous ligation dependent elongations unless another species of dimer is also present in the system. The explanation is given in Figure 5.

In summary there are the three principles that we used to limit the elongation catastrophe in this simple system.

1. Impossibility of formation of non-complementary h-bonded pairs.



**Figure 6:** A selection of unsuccessful iterations of the design, illustrating the ways in which various issues were solved. The designs are shown in chronological order. See text for details. Magnets are shown in red or blue depending on whether the north or south pole is oriented upwards. The weak bond magnets, whose orientation depends on the polymer type, are shown in white.

2. A high moment of inertia at the pivot point of a staggered end.
3. Improper alignment of p-bond passive hooks during an attempted templated ligation for N-mers where  $N > 2$ .

In combination these three factors significantly reduced the elongation catastrophe by limiting partial homologous ligation. The curved passive hooks previously described also helped by reducing the extent of non-catalysed ligation.

## A Phylogeny of Designs

A number of issues had to be solved simultaneously in order to produce a successful design. It took approximately 30 iterations to produce the final design, some of which can be seen in Figure 6. We have listed the issues that needed to be solved below.

i. The strong (p-)bonds must be unlikely to form spontaneously, i.e. the problem of reducing spontaneous generation.

ii. There must not be any reactions that catalyse p-bond formation, other than the intended template mechanism. For example, if two pairs of monomers joined by h-bonds come together, they must not line up at the right angle to form p-bonds.

iii. Once formed, the strong bonds must be strong enough that they rarely break. (In the final design they were strong enough not to break at all.)

iv. The strong bonds must form easily when catalysed by the weak bonds.

v. The weak (h-)bonds must form easily.

vi. The weak bonds must also break easily. This facilitates strand separation, as well as freeing up monomers that have become weak-bonded to other monomers, which would otherwise not be able to participate in catalysis.

vii. Once a dimer has catalysed the creation of another, the two 'strands' must be able to separate, i.e. the problem of product inhibition.

viii. The magnet in the weak bond must not be able to attach strongly to the magnet in the head or tail of another puck. Such unwanted bonds inhibit catalysis by occupying the bond points, and can also give rise to configurations that can catalyse the wrong type of dimer.

ix. The puck must be able to float effectively on the table. Designs with long thin protruding parts, or uneven weight distributions, can drag on the table's surface.

x. The pucks must not tend to jump off the table's surface and become stacked on top of one another. This tends to happen if two magnets with the same polarity are forced close to one another, or if the design features spikes that are too sharp.

Of these, issues i and ii were by far the hardest to solve. In most of our designs, including the final one, the strong bond works by requiring the two pucks to collide at a very precise angle. In many of the designs, if the collision occurred at a slightly different angle, a strong bond would often form anyway. This is because the head and tail magnets would tend to make the pucks slide into place to form a strong bond, or else the two pucks would sit together in a configuration where they could easily be nudged into the right position to form a strong bond. This was solved in the final design by adding long spikes to the strong bond constraints, in such a way that the magnets tend to pull the pucks away from, rather than towards, the strong bond configuration if the pucks are not correctly lined up. However, the pucks do still occasionally collide at the right angle to form a strong bond.

Since we could not substantially reduce the rate at which this occurs, we instead focused on increasing the rate of catalysis. We addressed issue iv by designing the weak bond to act as a pivot that guides the strong bonds into place.



Issue v was solved by making the weak bond protrude as much as possible from the body of the puck. This increases the range of relative angles at which two pucks can be oriented while still being able to form a weak bond. Issue vi was addressed by making the weak bond into a pivot that can swing fairly freely. As the joint hinges the two magnets are pushed further apart, so that the bond can break if it swings far enough. This could be fine-tuned by making very small changes to the magnets' positions. The "foot" mechanism was introduced to solve issue vii.

Issue viii was solved in the final design by the "spikes" in the head and tail sections (see Figure 3). These also help with issue i. The remaining issues were solved primarily by trial and error.

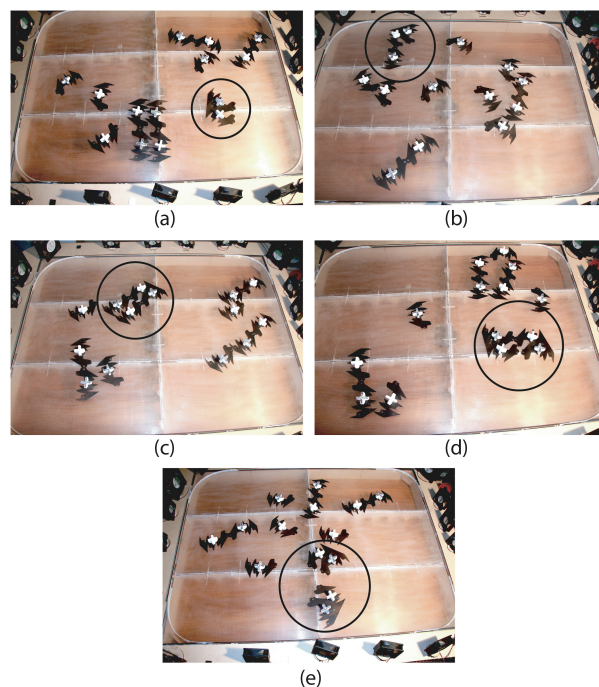
Figure 6 shows a selection of previous iterations of the design, illustrating some of these problems and how they were solved. Design (a) was ineffective because weak bonds formed only rarely. This is because the pucks have to be fairly specifically oriented with respect to one another in order for the weak bond magnets to come in range of each other. Additionally, the weak bond magnet of an 'A' type monomer can bond strongly to the tail magnet of another monomer, blocking catalysis. These two problems are solved in design (b) by making the weak bond protrude from the body of the puck, and by re-designing the strong bond so that the magnets are recessed away from the puck's edge. However, it is relatively easy for strong bonds to form spontaneously in this design, and they can also be catalysed by the edge of the table. The spikes added to the strong bonds in design (c) help to prevent spontaneous strong-bond formation, but they also interfere with the catalysis mechanism. This design also features a 'hump' on the opposite side to the weak bond; this is to prevent the edge of the table from catalysing bonds. Design (d) is the first to feature a weak bond that is designed to pivot around a particular point, with a correspondingly curved set of strong bond constraints. However, strong bonds can still form spontaneously quite easily, and weak bond formation is relatively rare.

Design (e) has a strong bond that is held together using repulsion rather than attraction (hence the head and tail magnets are of the same polarity). Unfortunately this tends to result in the magnets jumping off the table to stack on top of one another, since this is energetically preferable to being near one another in a repulsive configuration. The weak bonds have also been re-designed to be easier to form. Design (f) is similar but uses attracting magnets again; its main problems are that strand separation is very slow, and spontaneous strong bond formation is still an issue. Design (g) is the first to feature a mechanism to break one of the weak bonds when a strong bond is catalysed (two dimers cannot fit together in such a way that they are joined at both weak bonds). However, the spontaneous formation of strong bonds is still an issue, as is the formation of unwanted bonds between the weak and strong bond magnets. Design (h) uses Velcro rather than magnets for the strong bonds in an attempt to solve these issues. This idea was discarded because Velcro produces a loose joint, which means the strong bonds do not align accurately enough for catalysis to take place. However, we realised in testing this design that making the lock-and-key structures on the strong bonds wider helps to prevent spontaneous strong bond formation.

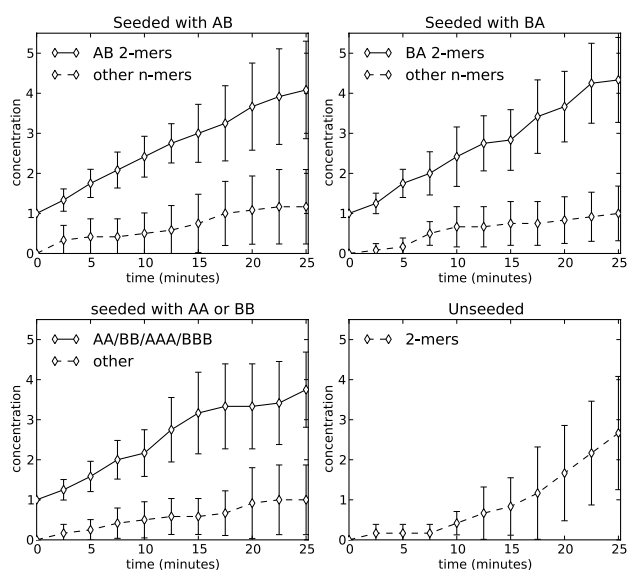
Design (i) is close to the final design and works fairly effectively. Its two remaining problems are that unwanted weak-strong bonds can form (although they are quite weak), and monomers can be attracted together by the strong bond magnets in such a way that a strong bond can form if they are nudged in the right way. These problems are solved in the final design by the addition of the central head and tail spikes (see figure 1), and by making the other spikes a lot larger.

We produced a total of 14 monomers, seven of type 'A' and seven of type 'B'. A total of 48 experiments were performed with the final design, each lasting 25 minutes. 36 of these were seeded trials, meaning that one dimer was added to a system containing the remaining 12 monomers. The system is allowed to run for a few minutes before adding the dimer, to ensure that the initial conditions do not affect the outcome. After the dimer was added we counted the number of each type of polymer every 2.5 minutes.

Of the 36 seeded trials, 12 were seeded with an 'AB' type dimer, 12 with type 'BA', 6 with type 'AA' and 6 with type 'BB'. Since 'AA' and 'BB' are two phases of the same



**Figure 7:** Photographs showing one round of the self-replication cycle. (a) An 'AB' dimer (circled) is placed into a system containing 6 'A' monomers (with white-topped sails) and 6 'B' monomers (black-topped sails). (b) A 'B' monomer joins to the 'A' part of the dimer via a weak bond. (c) An 'A' monomer joins via a weak bond to the 'B' part of the dimer, and its head constraints interlock slightly with the other monomer's tail constraints. (d) A collision with the table's edge or another molecule pushes the two monomers together, so that they form a strong bond. This breaks one of the two weak bonds. Note that both dimers are of type 'AB'. (e) Further collisions break the remaining weak bond, and the two strands separate. This completes the autocatalytic cycle.



**Figure 8:** Time series plots showing the results of letting the system run for 25 minutes, seeded with one dimer of a particular type, or with no dimer. In this plot, all polymers apart from those of the seed type are lumped into a single category. In the case of the trials seeded with AA or BB, we count AA, BB, AAA and BBB as a single category, since these can all be produced by the catalysis process from the seed type. Each plot shows the average over 12 trials. The error bars show a 95% confidence interval.

replicator, the latter two are plotted below as a single set of 12 trials.

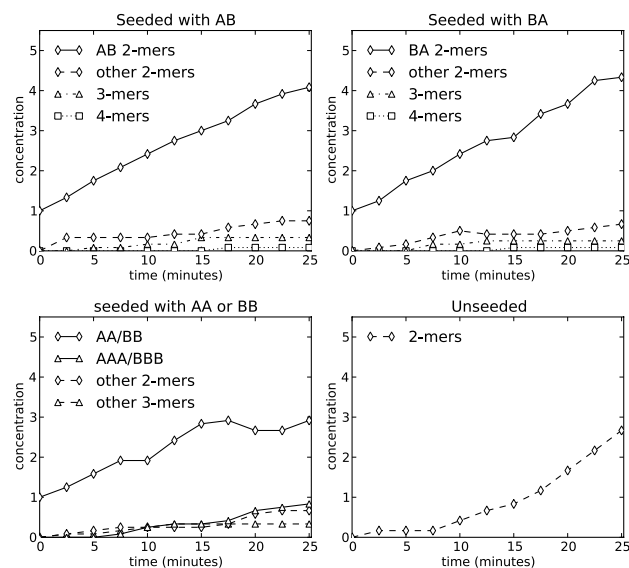
The control experiment involves initializing the system with seven 'A' type monomers and seven 'B' type ones, and is again run for 25 minutes. 12 such experiments were conducted.

## Results

The results are summarised below and in Figures 8 and 9. We count as a side reaction the production of any oligomer other than the seed type. In the AA/BB case we count AAA and BBB as copies of the original rather than as side-products, because there is no mechanism to prevent the formation of these 3-mers, and because they can still catalyse the production of new BB or AA dimers.

In 19 out of the 36 seeded trials, no side reactions took place during the 25 minutes of the trial. In these successful trials, an average of 4.3 duplicates (or, in the AA/BB case, elongations) of the seed were created in addition to the seed itself. The maximum possible number of copies is 6 in the AB or BA case, or 5 in the AA/BB case, with a miss-matched pair of monomers left over. This best-case performance was achieved in four of the trials.

In the remaining 17 seeded trials a side reaction produced an oligomer of a different species from the seed. In some trials this did not substantially disrupt the replication of the seed, but in others, particularly if the side reaction happened early in the trial, the side product produced more replicates than the seed dimer, effectively out-competing it by using up



**Figure 9:** Time series data from the same trials as Figure 8, with the reaction products split up by length. Note in particular the drop in concentration of AA and BB dimers towards the end of the trial as they are converted into AAA and BBB via elongation at staggered ends. (Error bars are omitted because they would be overlapping)

the monomer supply. Under some circumstances it is also possible for the side product to join to a dimer of the seed type in a staggered fashion (as in Figure 5), catalysing its elongation into a different species. For these reasons the mean number of duplicates of the seed after 25 minutes was only 1.7 in the 17 trials where side reactions occurred, or 3.1 over all 36 seeded trials. In four out of the 12 unseeded trials there were no side reactions, meaning that only monomers were present after 25 minutes. Over all 12 unseeded trials, an average of 2.7 oligomers were produced, of various species.

Time series data are shown in Figures 8 and 9, averaged over each of the four sets of 12 trials. In figure 8 all the side reaction products are lumped into a single category. The error bars show that the domination of duplicates of the seed over all other species is statistically significant to within a 95% confidence interval at every time step.

## Conclusions

The hexanucleotide replicator of von Kiedrowski was not evolvable because no mutant of the original sequence was capable of self-replication. Furthermore the ends of the molecules were blocked so that elongation was impossible. Breivik's model was not evolvable for the opposite reason; there was too much spontaneous generation and an elongation catastrophe. Here we have shown a way to achieve something in between, that at least has the potential for evolvability.

There is no doubt that the elongation catastrophe will be faced in all nanoscale self-replicating systems as well. Of-course, technology may allow such problems to be solved

somewhat trivially if monomers are allowed to contain switchable electromagnetic bonds (Groß, Küchler et al. 2009) and can implement a finite state machine (Griffith, Goldwater et al. 2005) thus avoiding issues of product inhibition and mismatching by simply allowing bonds to be arbitrarily made or formed based on perfect local information. However, this arbitrary programmability limits their utility in providing insight into possible molecular mechanisms of non-enzymatic template replication that depend on carefully evolved steric and force constraints, which is one of our main motivations here.

Of course, real molecular systems happen on vastly different spatial and temporal scales: our system has 14 monomers whereas a small chemical system might have  $10^{20}$ . In chemical systems interactions might occur only in a tiny majority of collisions, which we had to avoid in our experiments as it would have made the time scale too long. Nevertheless we believe the insights we have gained are useful.

The implication for the origin of life is that it is possible to produce monomers that self-limit to some extent the lengths of strands that can be self-assembled according to the mechanisms shown in Figure 5. It may be the case that such primitive methods may have been among the first evolved to combat the elongation catastrophe. The production of this physical model has (at least for us) been helpful as E.F. Moore said it would be.

## Acknowledgements

The work was funded by the E-FLUX FP7 EU Grant. Many thanks to Simon McGregor, Chris Buckley and Eors Szathmáry for critical discussion. Roderich Gross and Francesco Mondada were instrumental in suggesting to us the possibility of using air-hockey and fans for creating a stochastic environment. Jarle Breivik and Guenter von Kiedrowski's work was the great inspiration for this project.

## References

- Breivik, J. (2001). "Self-organization of template-replicating polymers and the spontaneous rise of genetic information." *Entropy* 3: 273-279.
- Eigen, M. (1971). "Selforganization of matter and the evolution of biological macromolecules." *Naturwissenschaften* 58(10): 465-523.
- Fernando, C. T., G. von Kiedrowski, et al. (2007). "A Stochastic Model of Nonenzymatic Nucleic Acid Replication: "Elongators" Sequester Replicators." *Journal of Molecular Evolution* 64: 572-585.
- Fodor, J. A. and Z. W. Pylyshyn (1988). "Connectionism and cognitive architecture: A critical analysis." *Cognition* 28: 3-71.
- Griffith, S., D. Goldwater, et al. (2005). "Self-replication from random parts." *Nature* 437: 636.
- Groß, R. M., S., L. Küchler, et al. (2009). Towards an Autonomous Evolution of Non-Biological Physical Organisms. *Proceedings of the 10th European Conference on Artificial Life, In Lecture Notes in Computer Science*, Berlin, Germany, 2009. Springer Verlag.

- Moore, E. F. (1962). *Machine models of self-reproduction*. Proceedings of the 14th Symposium in Applied Mathematics, American Mathematical Society, New York.
- Penrose, L. S. (1958). "Mechanics of Self-reproduction." *Ann. of Human Genetics* 23: 59-72.
- Penrose, L. S. (1959). "Self-reproducing Machines." *Scientific American* 200: 105-114.
- Penrose, L. S. and R. Penrose (1957). "A self-reproducing Analogue." *Nature* 179: 1183.
- Price, G. R. (1970). "Selection and covariance." *Nature* 227: 520-521.
- Szathmáry, E. (2000). "The evolution of replicators." *Phil. Trans. Roy. Soc. Lond. B* 355: 1669-1676.
- Szathmáry, E. (2006). "The origin of replicators and reproducers." *Philos. Trans. R. Soc. London. B. Biol. Sci.* 361(1474): 1761-1776.
- Szathmáry, E. and J. Maynard Smith (1997). "From replicators to reproducers: the first major transitions leading to life." *Journal of Theoretical Biology* 187(4): 555-571.
- Von Kiedrowski, G. (1986). "A self-replicating hexadeoxy nucleotide." *Angew. Chem. Int. Ed. Engl.* 25: 932-935.

# Clustering Behavior of a Bio-inspired Decentralized Aggregation Scheme

Nikolaos Vlassopoulos<sup>1</sup> and Nazim Fatès<sup>2</sup>

INRIA Nancy — Grand Est / LORIA

MAIA Team

<sup>1</sup>Nikolaos.Vlassopoulos@inria.fr, <sup>2</sup>Nazim.Fates@inria.fr

## Abstract

This note reviews a bio-inspired scheme for aggregating autonomous agents in the absence of global communication or coordination, a problem that is known as *Decentralized Gathering*. We present results on the clustering behavior of the agents, as we vary the main parameter that controls the agents' aggregation. Our observations show that there exist two phenomenologically different behaviors, characterized by two different evolutions of the number of clusters with time. We relate these different behaviors to the coupling of two factors: a change in the scale of the interaction range of the agents and a change in the significance of the local fluctuations in the model.

## Introduction

Assume that a large number of autonomous and *identical* agents are scattered on a plane, and that there is no global authority to coordinate their actions nor any means of global communication. The problem of gathering these agents in a small area is known as the Decentralized Gathering. This problem is known to be difficult in general and is even be impossible to solve exactly in some continuous space frameworks (Prencipe (2007)).

One approach to solving the decentralized gathering problem consists of imitating the behavior of the amoebae species *Dictyostelium discoideum* (Fatès (2010)). The main characteristic of this approach is the existence of an *active* environment that conveys simple messages among the agents, which are called *virtual amoebae*. The agents interact with the environment by either initiating the transmission of a message or by detecting the existence of messages in their local environment. These two types of interaction are the building components for a stigmergic behavior.

The virtual amoebae aggregation scheme has been shown to exhibit a rich dynamical behavior (Fatès (2010); Vlassopoulos and Fatès (2010)) and to be robust. In this note, we focus on a Cellular Automaton-based (CA) instance of the aggregation scheme, as it has been described in (Fatès, 2010), and present a qualitative description of the two contrasting, clustering behaviors that can be observed in the model. As we will describe, these behaviors result from a

change of scale on the interactions among the agents, from short-ranged to long-ranged. Interestingly, the aggregation behavior persists despite this change of scale.

## Virtual Amoebae Aggregation Scheme

### Active Environment

The existence of an active environment simplifies the agent behavior, by delegating parts of its complexity to the environment and allows for “self-sustained” messages, that can travel arbitrary large distances. The active environment is modeled with a two-dimensional Greenberg Hastings reaction-diffusion cellular automaton (GHCA, see Greenberg et al. (1978)). The CA consists of an array of cells of dimensions  $L \times L$ , a set of cell states,  $\Sigma$ , a set of transition rules for the states and, for each cell, a set of cells that constitute its neighborhood  $\mathcal{N}_c$ .

In the GHCA,  $\Sigma = \{M, \dots, 0\}$ , where  $M$  is called the *excited* state,  $M - 1, \dots, 1$  are called the *refractory* states and 0 is the *neutral* state. A cell becomes excited only if it is neutral and if at least one of its neighboring cells is excited. An excited cell will become refractory in the next time step and then decrease its state until it reaches the neutral state. The dynamics of the GHCA involve “waves” composed of wavefronts of excited cells followed by refractory cells that extend outwards from an excitation. Most importantly, when two reaction-diffusion wavefronts meet, they annihilate.

### Agents

For simplicity, we consider agents as particles that can read the states of the cell on which they reside as well as the state of the neighboring cells. The virtual amoebae behavior is then summarized as follows: If the state of the cell where an agent resides is 0 (neutral), then, at each time step, the agent initiates (“fires”) a reaction-diffusion process with probability  $\lambda$ , by setting the state of the cell to  $M$ . If the cell is neutral and an excited neighbor is detected, the agent moves towards the excited neighbor, choosing randomly if more than one are excited. Otherwise, if the cell is in a refractory state, do nothing. Here,  $\lambda$ , the *firing rate*, is the most important parameter of the aggregation model. In our study, each cell



can hold at most two agents. Increasing the cell capacity affects mainly the spatial dimensions of the clusters and, for high values of  $\lambda$ , the aggregation time. Figure 1 shows the aggregation process for two different values of  $\lambda$ .

One may thus wonder what is causing the agents to aggregate in both cases, where we see a completely different quantitative behavior. A partial answer to this question is: the presence of fluctuations, both in terms of the density and in terms of emission of reaction diffusion waves.

## Clustering Behavior

In a previous work (Vlassopoulos and Fatès (2010)) we have shown that there exists an optimal value of  $\lambda$  such that the aggregation time is minimized. The two different clustering behaviors became apparent while studying the aggregation properties of the model (Fig. 1). In both cases, the agents, given a sufficient amount of time, will aggregate to a single cluster, but as we can observe, this is accomplished by exhibiting two completely different sequences of intermediate macroscopic configurations. In the second (bottom) figure, where  $\lambda$  is large enough, we observe that the agents form small clusters that progressively merge into bigger ones. This process continues until there are a few large clusters that persist for a relatively large amount of time before merging into a single one. However, in the first (top) figure we can observe that the agents “collapse” into a single cluster, without going through the formation of intermediate, persisting, clusters.

From our experiments so far, we have observed that this transition, i.e. from (A) the formation of competing clusters and progressive merging into a single one to (B) the “collapse” of the agents into a single cluster, and inversely, appears to be continuous. One important remark is that high values of  $\lambda$  favor small-range interactions among the agents, in the sense that the distance a wave will manage to travel, and consequently, the number of agents it will manage to interact with, before it is annihilated by the presence of other waves in the environment decreases as  $\lambda$  increases. Accordingly, small values of  $\lambda$  will allow a wave to travel larger distances and interact with more agents before it is annihilated, and therefore can be considered as larger-range interactions. To sum up, the aggregation process persists, in spite of the scale changes and the different macroscopic behavior that results from these changes.

## Fluctuations as a Source of Order?

What is the “driving force” of aggregation in the different scales we described? The common denominators that “destabilize” persisting clusters and cause agents to collapse in both behaviors are the fluctuations, in terms of density and in terms of emissions. The density fluctuations exist even for very small values of  $\lambda$ , but in this case, where the interaction wavelengths are greater than the grid size, they are not significant and the agents aggregate into one cluster. For

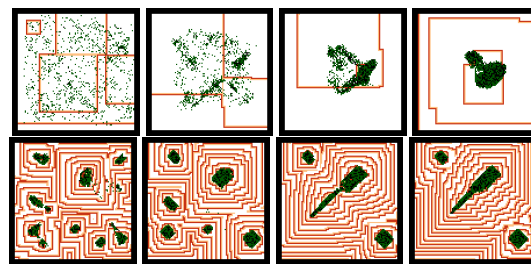


Figure 1: Aggregation instances for different values of  $\lambda$ . Top:  $\lambda = 1 \cdot 10^{-5}$  Bottom:  $\lambda = 8 \cdot 10^{-2}$ . Agents are shown with green and the reaction diffusion wavefronts with orange. The initial number of agents is 400.

high values of  $\lambda$  they become important and are the main reason for the generation of the initial small local clusters that will subsequently merge, until only one cluster remains, but also one of the reasons that cause cluster to merge, since that out of two clusters with (sufficiently) different number of amoebae, we expect that the larger one will emit more waves, in average. The fluctuations in the emission times are the driving force that causes the clusters to merge, for both small and large values of  $\lambda$ . However, it is interesting that we observe the same effective behavior of the system in different scales. More precisely, the same “forces” that cause two amoebae to merge into a cluster, will cause two clusters to merge into a larger one and so on, until only one cluster exists. The merging behavior seems to be similar at different scales, which leaves us with the question: are there quantities that are invariant with respect to rescaling?

To conclude, we described a bio-inspired model that shows how it is possible to exploit the presence of fluctuations in a constructive way, in order to drive a system to a desired final state. The existence of an active environment simplifies the overall model design, but also increases the significance of fluctuations, that constitute a major factor to the operation and robustness of the model.

## References

- Fatès, N. (2010). Solving the decentralised gathering problem with a reaction-diffusion-chemotaxis scheme - social amoebae as a source of inspiration. *Swarm Intelligence*, 4(2):91–115.
- Greenberg, J. M., Hassard, B. D., and Hastings, S. P. (1978). Pattern formation and periodic structures in systems modeled by reaction-diffusion equations. *Bulletin of the American Mathematical Society*, 84(6):1296–1327.
- Keller, E. and Segel, L. (1970). Initiation of slime mold aggregation viewed as instability. *Journal of Theoretical Biology*, 26:399–415.
- Prencipe, G. (2007). Impossibility of gathering by a set of autonomous mobile robots. *Theor. Comput. Sci.*, 384(2-3):222–231.
- Vlassopoulos, N. and Fatès, N. (2010). How fast can Virtual Amoebae aggregate? Technical report, INRIA - LORIA. <http://hal.archives-ouvertes.fr/inria-00462156/fr/>.

# Measuring Information Storage and Transfer in Swarms

X. Rosalind Wang<sup>1</sup>, Jennifer M. Miller<sup>2</sup>, Joseph T. Lizier<sup>3</sup>, Mikhail Prokopenko<sup>1</sup> and Louis F. Rossi<sup>2</sup>

<sup>1</sup>CSIRO Information and Communications Technology Centre, PO Box 76, Epping, NSW 1710, Australia

<sup>2</sup>Department of Mathematical Sciences, University of Delaware, Newark DE 19716, USA

<sup>3</sup>Max Planck Institute for Mathematics in the Sciences, Inselstraße 22, 04103 Leipzig, Germany  
Rosalind.Wang@csiro.au

## Abstract

Spatial aggregation of animal groups give individuals many benefits that they would not be able to obtain otherwise. One of the key questions in the study of these animal groups, or “swarms”, concerns the way in which information is propagated through the group. In this paper, we examine this propagation using an information-theoretic framework of distributed computation. Swarm dynamics is interpreted as a type of distributed computation. Two localized information-theoretic measures (active information storage and transfer entropy) are adapted to the task of tracing the information dynamics in a kinematic context. The observed types of swarm dynamics, as well as transitions among these types, are shown to coincide with well-marked local and global optima of the proposed measures. Specifically, active information storage tends to maximize as the swarm is becoming less fragmented and the kinematic history begins to strongly inform an observer about the next state. The peak of transfer entropy is observed to appear at the final stages of merging of swarm fragments, near the “edge of chaos” where the system actively computes its next stable configuration. Both measures tend to minimize for either unstable or static swarm configurations. The results here show these measures can be applied to non-trivial models, most importantly, they can tell us about the dynamics within these model where we can not rely on visual intuitions.

## Introduction

There are many examples of spatial aggregation in animal groups in nature including schools of fish, swarms of locusts, herds of wildebeest, and flocks of birds (Lissaman and Shollenberger, 1970; Parrish and Edelstein-Keshet, 1999; Sinclair and Norton-Griffiths, 1979; Uvarov, 1928). Such aggregations give group members the benefit of protection, mate choice, and information that an individual might not be aware of on its own such as the location of a food source, predator, or migratory route (Camazine et al., 2003; Partridge, 1982). There is considerable evidence in many species that individuals can only perceive their neighbors rather than the entire group (Camazine et al., 2003). Typically, these groups are referred to as self-organized since they form without any centralized control.

In self-organized groups, complex large-scale patterns and structures emerge through individual decisions based on

perception of local conditions. For example, in response to a predator, many schools of fish display complex collective patterns of motion, including compression, vacuole, flash expansion, milling, or form highly parallel translating groups (Parrish et al., 2002).

The key questions in the study of swarms and “swarm intelligence” concern how the local interactions map to the large-scale behavior. Finding answers to some of these questions has broad implications in ecological and artificial systems. The way in which information is propagated in animal groups is poorly understood (Couzin et al., 2006). Recently, there has been some effort to understand this transfer of information. Couzin et al. (2006) depict schooling of fish as a type of distributed information processing. The authors refer to the collective memory of the school, and describe a wave of turning (triggered by a small number of fish reacting to some sensory information) as “information cascades” spreading across the school. The authors comment that such mechanisms can transmit information at a speed faster than that of an incoming predator, with such computational capability providing an evolutionary advantage.

Conjectures also relate known phase transitions in flocking or schooling behavior to the underlying information dynamics of the computations they are carrying out. Miramontes (1995) describes phase transitions in the maximization of activation levels in ant foraging behavior with respect to ant density, and suggests that this is reflected in maximization of information transfer between the ants. In a similar vein, Couzin (2007) describes the phase transition of effective flocking behavior occurring only at intermediate sensory ranges between individual agents in terms of the capacity for information transfer that the sensory range allows: too short a sensory range does not allow enough information transfer to form cohesive groups; too large a range permits rampant spreading of irrelevant information which erodes group cohesion. These notions are very similar to the generic descriptions of the information dynamics of order-chaos phase transitions under the “edge of chaos” hypothesis (e.g. see Langton (1990)).

In this paper, we examine the propagation of information

in swarms using a recent framework that characterizes the information dynamics of distributed computation in terms of the elements of Turing universal computation (Lizier et al., 2008b, 2010, 2011, 2007). In particular we seek to measure how much information is stored, and how much information is transferred to and by each agent in the swarm at each time step. The information dynamics of distributed computation have recently emerged as an important tool for studying complex systems, such as cellular automata (CAs) (Lizier et al., 2007, 2008b) and random Boolean networks (RBNs) (Lizier et al., 2008a). This approach has quantitatively identified the coherent structures known as “gliders” as the dominant information transfer agents in CAs.

We note that the dynamics of animal groups can be seen as a type of distributed computation. As we will show, at each time step, each agent “computes” its new state as a function of its previous state and the relative states of each of its neighbors. In this way, information relevant to that computation can be stored in regular patterns of behavior, transferred from the relative state of each neighbor, and processed when an agent combines the effects of multiple sources together.

We begin this paper with an overview of the swarm model and the information dynamics framework. This is followed by discussion on how we applied the information dynamics framework to swarms, introducing specific techniques to handle the *amorphous* computational structure and measure state transitions using relative variables. Finally, the results are shown and discussed.

### Three Zone Model for Swarms

There are two methodologies for modeling and simulating aggregations of discrete individuals. Individual-based or agent-based models are discrete-time models that update individual positions and velocities based on positions and velocities at the previous discrete time-step. Particle models capture swarm dynamics as a coupled system of ordinary differential equations. Individual-based and particle models can be nearly equivalent if they are designed consistently and if the step size of the individual-based model is taken to be small. Regardless of the methodology, the interactions between individuals drive the dynamics of the model. A common behavioral model for swarms is to have each individual respond to neighbors in three concentric zones that are used to define the behavioral rules of motion (Aoki, 1982; Couzin et al., 2002; Huth and Wissel, 1992; Lukeman et al., 2010; Vicsek et al., 1995). The individual responds differently to neighbors in each of the three zones through repulsion, orientation, or attraction, respectively. An individual moves away from neighbors in the zone of repulsion, aligns its velocity with that of neighbors in the zone of orientation, and moves toward neighbors in the zone of attraction.

In the models featured in this study, we use continuous kernels to define the zones which describe individual behavior with smooth transitions from one type of movement

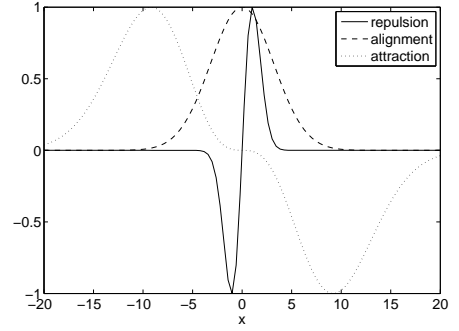


Figure 1: Normalized amplitudes of the social interaction kernels.

to another (see (Miller et al., 2011)). One of the advantages of this model is that it can be scaled up to a continuum limit where there are infinitely many individuals expressed through a density function and a velocity field. The dynamics of the system is governed by a system of partial differential equations which we can analyze using tools that are not available to discrete models. While our kernels do not have compact support, exponential decay guarantees that individual behavior is dominated by nearby neighbors; similarly, interactions with distant members is exponentially small. The three responses are combined to determine the desired velocity  $\vec{v}_d$ . For a reference individual  $i$ , we define the displacement vector  $\vec{s}_{ij} = \vec{s}_j - \vec{s}_i$ , where  $\vec{s}$  is the individual’s position. The behavioral input vector is then  $\vec{v}_d = \vec{v}_{r,i} + \vec{v}_{o,i} + c_a \vec{v}_{a,i}$ :

$$\vec{v}_{r,i} = \sum_{j=1}^N -\frac{1}{8\pi\sigma_1^4} \vec{s}_{ij} \exp(-|\vec{s}_{ij}|^2/4\sigma_1^2), \quad (1)$$

$$\vec{v}_{o,i} = \frac{\sum_{j=1}^N \frac{1}{4\pi\sigma_2^2} \exp(-|\vec{s}_{ij}|^2/4\sigma_2^2) \vec{v}_j}{\sum_{j=1}^N \frac{1}{4\pi\sigma_2^2} \exp(-|\vec{s}_{ij}|^2/4\sigma_2^2)}, \quad (2)$$

$$\vec{v}_{a,i} = \sum_{j=1}^N \frac{1}{64\pi\sigma_3^6} \vec{s}_{ij} |\vec{s}_{ij}|^2 \exp(-|\vec{s}_{ij}|^2/4\sigma_3^2), \quad (3)$$

where  $c_a$  specifies the relative importance of attraction to orientation and repulsion and  $\sigma_k$ ,  $k = \{1, 2, 3\}$ , are constants that control the sizes of the zones. A cross section of the amplitude of these kernels is shown in Figure 1. The behavioral input vector is used in different ways to control the change in velocity. Once the velocity has been determined, the positions are updated accordingly.

We examined the information transfer in two different second order models. Our variable speed model updates the velocity by setting

$$\vec{v}_i^{n+1} = \vec{v}_i^n + \delta\tau \cdot \kappa (\vec{v}_{d,i}^n - \vec{v}_i^n) \quad (4)$$

where  $\delta\tau$  is the time step length and  $\kappa$  is the turning rate. The constant (unit) speed model updates the direction  $\theta_i$  by

$$\theta_i^{n+1} = \theta_i^n + \delta\tau \cdot \kappa (\vec{v}_i^n)^\perp \cdot \vec{v}_{d,i}^n, \quad (5)$$

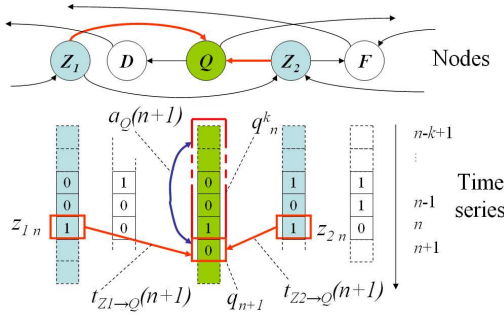


Figure 2: Information dynamics in a distributed network. For node  $Q$ , this figure displays the local active information  $a_Q(n+1, k)$  and the local transfer entropies  $t_{Z_1 \rightarrow Q}(n+1)$  and  $t_{Z_2 \rightarrow Q}(n+1)$  from each of the causal information sources  $\mathbf{Z}_Q \in \{Z_1, Z_2\}$  at time  $n+1$ .

where

$$(\vec{v}_i^n) = [\cos \theta_i^n, \sin \theta_i^n]^T, \quad (6)$$

$$(\vec{v}_i^n)^\perp = [-\sin \theta_i^n, \cos \theta_i^n]^T. \quad (7)$$

It has been shown that for  $c_a > 1$ , uniform infinite swarms are linearly unstable (Miller et al., 2011). Finite groups that are initialized on a square lattice reorganize into an attractor when  $c_a > 1$ . For the variable-speed model, this attractor is circular with a variable density. In the constant-speed model, the attractor is anisotropic as well as variable density.

### Information Dynamics

Information theory (MacKay, 2003) has been well used in complex systems (Langton, 1990; Miramontes, 1995; Schreiber, 2000), and is the natural domain to look for a framework to describe the information dynamics here. Lizier et al. (2007, 2008b, 2010, 2011) have proposed such a framework for local information dynamics, describing distributed computation in terms of information storage, transfer and modification at each spatiotemporal point in a complex system. In this paper, we focus on the first two terms.

The **information storage** of an agent in the system is the amount of information in its past that is relevant to predicting its future. We will compute the *active information storage* (AIS) component, which is the stored information that is currently in use in computing the next state of the agent (Lizier et al., 2011, 2007). We focus on the active information since it yields an immediate contrast in the relative contributions of storage and transfer to each computation. As shown in Fig. 2, the *local active information storage* for agent  $Q$  is defined as the local (or unaveraged) mutual information between its semi-infinite past  $q_n^{(k)}$  (as  $k \rightarrow \infty$ ) and its next state  $q_{n+1}$  at time step  $n+1$ :

$$a_Q(n+1) = \lim_{k \rightarrow \infty} \log_2 \frac{p(q_n^{(k)}, q_{n+1})}{p(q_n^{(k)})p(q_{n+1})}, \quad (8)$$

with  $a_Q(n, k)$  representing an approximation with finite history length  $k$ . The *active information* is the average over time (or equivalently weighted by the distribution of  $(q_n^{(k)}, q_{n+1})$ ):  $A_Q(k) = \langle a_Q(n, k) \rangle$ . From our computational perspective, an agent can store information regardless of whether it is causally connected with itself. This is because information storage can be facilitated in a distributed fashion via one's neighbors (Lizier et al., 2011).

The **information transfer** between a source and a destination agent is defined as the information provided by the source about the destination's next state that was not contained in the past of the destination. The information transfer is formulated in the *transfer entropy* (TE), introduced by Schreiber (2000) to address concerns that the mutual information (as a de facto measure of information transfer) was a symmetric measure of statically shared information. The *local transfer entropy* (Lizier et al., 2008b) from a source agent  $Z$  to a destination agent  $Q$  is the local mutual information between the previous state of the source  $z_n$  and the next state of the destination  $q_{n+1}$ , *conditioned* on the semi-infinite past of the destination  $q_n^{(k)}$  (as  $k \rightarrow \infty$ ):

$$t_{Z \rightarrow Q}(n+1) = \lim_{k \rightarrow \infty} \log_2 \frac{p(q_{n+1}|q_n^{(k)}, z_n)}{p(q_{n+1}|q_n^{(k)})}. \quad (9)$$

Again,  $t_{Z \rightarrow Q}(n, k)$  represents finite- $k$  approximation, and the *transfer entropy* is the (time or distribution) average:  $T_{Z \rightarrow Q}(k) = \langle t_{Z \rightarrow Q}(n, k) \rangle$ . A schema representation of the process to compute the local transfer entropy is shown in Fig. 2. Importantly, the transfer entropy properly measures a directed, dynamic flow of information, unlike mutual information measures which measure correlations only. Note that one can also condition the TE on another information contributor  $W$  to form the *conditional transfer entropy* (Lizier et al., 2010):

$$t_{Z \rightarrow Q|W}(n+1) = \lim_{k \rightarrow \infty} \log_2 \frac{p(q_{n+1}|q_n^{(k)}, w_n, z_n)}{p(q_{n+1}|q_n^{(k)}, w_n)}. \quad (10)$$

### Information Dynamics in Swarms

To apply the information dynamics framework to swarm models, we will need to make two important modifications to the use of Eq. (8) and Eq. (9).

**Accumulation of observations across agents:** In CAs, the probability distribution functions in Eq. (9) for the transfer entropy from agent  $Z_i$  to  $Q_{i+1}$  are estimated over observations from all agent pairs for the corresponding transfer channel (e.g. across one cell to the right per unit time step): the agents there were completely *homogeneous* in connectivity pattern and function. In RBNs, the probability distribution functions of Eq. (9) were estimated at single causal pairs  $Q$  and  $Z$  only, since the agents were *heterogeneous* in connectivity pattern and function.

In contrast to both of these applications, swarm computation is *amorphous*, with neither homogeneous computational structure across agents, nor with fixed computational relationships between heterogeneous causal pairs. That is, the causal relationships between agents change too rapidly to reliably estimate the probability distribution functions on any transient single causal pair. A pair of particles,  $p_1$  and  $p_2$ , could be close enough for a causal interaction at one time step, but move outside the interaction zones at next step. Therefore, calculating information transfer between single causal pairs over all time would not give us a good representation of the actual information within the system.

Helpfully, swarm models exhibit homogeneously *functional* agents, and so we can accumulate observations for the probability distribution functions in Eq. (9) from *every* transient causal interaction. That is, one adds individual causal interactions between many different agent pairs to the set of observations, without requiring any one of those agent pairs to maintain a causal relationship over all time steps. When one particle  $p_i$  is within proximity to have a causal effect over another particle  $p_j$ , their interaction is counted for the information-theoretic probability distribution functions, but when  $p_i$  is outside causal range of  $p_j$ , no observation is recorded. These probability distribution functions can then be applied to compute the local apparent transfer entropy between two different particles  $p_k$  and  $p_l$  which are causally connected at a different time step, because of the *homogeneous* nature of the *functionality* of each agent.

#### Measuring state transitions with relative variables:

Continuing the idea of accumulating observations over comparable interactions, we note that if two pairs of particles have the exact same *relative* positions and velocities, but have different *absolute* positions and headings, then the information dynamics of the two pairs should be the same. As such, we will focus our measurements on the computation of the *change* in velocity of the destination agent at each time step. Not only does this remain in the spirit of the original formulation of the transfer entropy by Schreiber (2000) (considering how much information a source adds about the state *transition* of the destination), but focuses directly on the causal relationship between particles (since a velocity change is computed rather than an absolute velocity).

Let  $p'$  be a particle that is within another particle  $p$ 's zones of interaction, so  $p$  and  $p'$  form a causal pair. Our aim is to find what influence does  $p'$  have on  $p$ . To reiterate,  $\vec{s}^n$  is the position and  $\vec{v}^n$  is the velocity at time  $n$ . The relevant variables for the conditional transfer entropy in Eq. (10) are:

$$z_n = \{\vec{s}_p^n - \vec{s}_{p'}^n, \vec{v}_p^n - \vec{v}_{p'}^n\}, \quad (11)$$

$$w_n = |\vec{v}|^n, \quad (12)$$

$$q_n^{(k)} = \vec{v}_p^n - \vec{v}_p^{n-1}, \quad (13)$$

$$q_{n+1} = \vec{v}_p^{n+1} - \vec{v}_p^n. \quad (14)$$

In other words:

- the source variable is the relative positions and velocities between the particles at time step  $n$ ;
- we condition the transfer on the speed of  $p$ ,  $|\vec{v}|^n$  at  $n$ ;
- the past state of the destination is the change in velocity of  $p$  at time step  $n$ ; this means we use  $k = 1$  due to the finite number of observations;
- and the destination variable next state is the change in velocity of  $p$  at  $n + 1$ .

Note that the destination past, next state and conditioned variables contain all of the relevant information about the state update  $q_{n+1}|q_n^{(k)}, w_n$  of the destination variable, from the perspective (or relative frame of reference) of  $p$  itself. The relative positional information at any time step (including the next step  $n + 1$ ) is then obtainable using the change in velocity terms. Given these destination variables, note that the source variable captures all relevant information from  $p'$  about the state transition of the destination.

Importantly,  $|\vec{v}|^n$  is included here since the absolute velocity of the particle may have some indirect influence on the change in state (e.g. by influencing how often the source and destination have recently interacted). We could additionally include  $|\vec{v}|^{n+1}$  in the next state  $q_{n+1}$  (then absorbing  $|\vec{v}|^n$  into  $q_n^{(k)}$  also to make a calculation of an ordinary transfer entropy here); however this ties the tuple  $(z_n, q_n^{(k)}, q_{n+1})$  to the absolute heading of  $p$ , removing the advantage of accumulating observations over comparable interactions that we had gained from the use of relative variables.

In this manner, the local conditional transfer entropy can be estimated for each transient causal relationship. Similarly, the local active information can be estimated from the destination and history of destination variables of each particle interaction. Agents in a swarm can be seen to use *stored* information when their behaviour is predictable from their own past (in isolation from other agents), and to *transfer* information in the way that their relative positions and velocities influence changes in velocity in other agents.

## Results

We applied the framework to swarm simulation with the particles initially in 3 squares in a checker configuration. Each initial square has  $28 \times 28$  particles, and thus we have a total of 2352 particles in the system. Fig. 8(a) shows a configuration close to the swarm's initial positions.

As discussed in earlier section, two different second order swarm models were used: variable speed and constant speed. As discussed earlier, for  $c_a > 1$ , swarms evolve into coherent attractors, here we set  $c_a = 5$  (Miller et al., 2011). We run the models until the swarms reach a steady state, that is, the shapes of the swarms do not change much. At this point, information transfer approaches zero and the computation can be seen as complete. Measuring the information dynamics during this transient period means that we are studying the computation of the swarm's steady state.

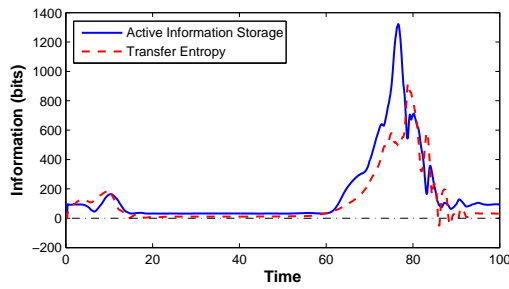


Figure 3: Total active information storage and information transfer over time for variable speed model. The values are averaged over 1000 repeated experiments, where each experiment randomly picked around 0.005% of the total interactions.

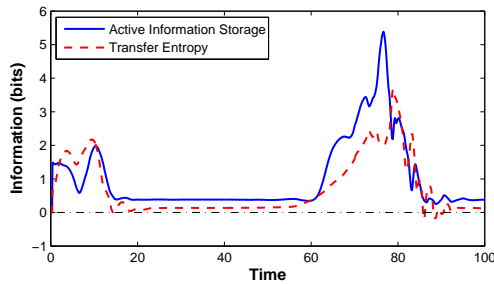


Figure 4: Average active information storage per particle and average information transfer per particle pair interaction over time for variable speed model.

The swarms reach a steady state at the time  $\tau < 100$  for variable speed model and  $\tau < 150$  for constant speed model. We used a step size of  $\delta\tau = 0.1$  (sufficient for resolving the dynamics of the swarms) and hence gather data for 1000 steps for variable speed model, and 1500 steps for constant speed model. At every step, each particle has on average several hundred neighbors within its zones of interaction, this means the total number of interactions is in the order of  $1 \times 10^9$ , which is much too large for the kernel estimation of  $p$ . Therefore, we randomly picked a fraction of the interactions at each time step for the calculation. We used a frequency of 20000, randomly picking 0.005% of the total interactions at each time step. This gives us approximately  $1 \times 10^5$  data points, thus we used a kernel width of 0.23 for kernel estimation with a normalized kernel. This is repeated 1000 times and the results averaged to give a better representation of the total interactions.

### Second order, variable speed model

Fig. 3 shows the total information storage and transfer at each time step over all time, e.g.  $\sum_Q a_Q(n, k)$ . Note that this is the average total over 1000 repeated experiments, where each experiment randomly picks 1 in 20000 interactions for the calculation. Therefore, the actual value should be 4 orders of magnitude larger than the values shown in this figure. Further, while the transfer values are summed over

all interactions, the storage values are summed over all  $p$ 's in the interactions with each  $p$  counted once.

Fig. 4 shows the average storage per particle and average transfer per interaction at each time step over time. Comparing this with the total information per time step, we can see that the second half of Fig. 4 has smaller magnitude relative to the first half. This shows that for  $\tau > 60$  the number of interactions in the swarm increased. However, the *shapes* of the plots do not differ from those in Fig. 3, which means the change in information dynamics values were not simply due to the change in the number of interactions.

In Fig. 3 we can notice three distinct epochs in the information dynamics of the swarm: (1) between  $\tau = 0$  and  $\sim 20$  where there is a small local maxima in the information dynamics; (2) between  $\tau \sim 20$  and  $\sim 60$  where the values remain low and constant; and finally (3) between  $\tau \sim 60$  and 100 where there were large changes in the values.

Fig. 5 shows some snapshots of key steps during the swarm simulation as identified by Fig. 3. Comparing the two, we can see that the first epoch corresponds to when the three initial squares 'collapse' to form three discs (Fig. 5(a)&(b)); epoch 2 corresponds to the three discs moving but not interacting with each other (Fig. 5(c)); and the final epoch is when the three discs come into contact with each other to form a single swarm (Fig. 5(d)-(j)).

Epoch 3 is the most interesting for the swarm simulation in terms of the resulting information dynamics within the system. Both information values start increasing at around  $\tau = 59.5$  (Fig. 5(d)) when the three separate groups become close enough for the outer particles of each to affect and be affected by those in the other groups.

As the groups merge, the two outer groups move towards the middle group, squeezing the middle one and triggering increases in both storage and transfer. This behavior continued until  $\tau = 75.1$  when the middle group was squeezed to the smallest size it can stand as seen Fig. 5(e), when transfer reaches a local maximum. High local transfer shows that the source is informative about the next state of the destination.

Between  $\tau = 75.1$  and 76.6 (Fig. 5(e)&(f)), the swarm tried to reorganize itself to a more stable configuration. This can be seen from the drop in transfer but rapid increase in storage values until at  $\tau = 76.6$  when the values reach local minimum or global maximum, respectively. High local storage shows the history strongly informs an observer about the next state. Thus, the increase in average storage value per particle as shown in Fig. 4 shows that the particles' velocities are increasingly predictable from their history.

Between  $\tau = 76.7$  and 78.9 (Fig. 5(f)&(g)), the swarm finally merged into one group. Fig. 3 and 4 shows that during this time information storage decreased while transfer increased, indicating a transition to another (more stable) swarm configuration. At  $\tau = 78.9$ , the storage value reaches a local minimum and the transfer value reaches the global maximum. The maximum transfer entropy value shows that



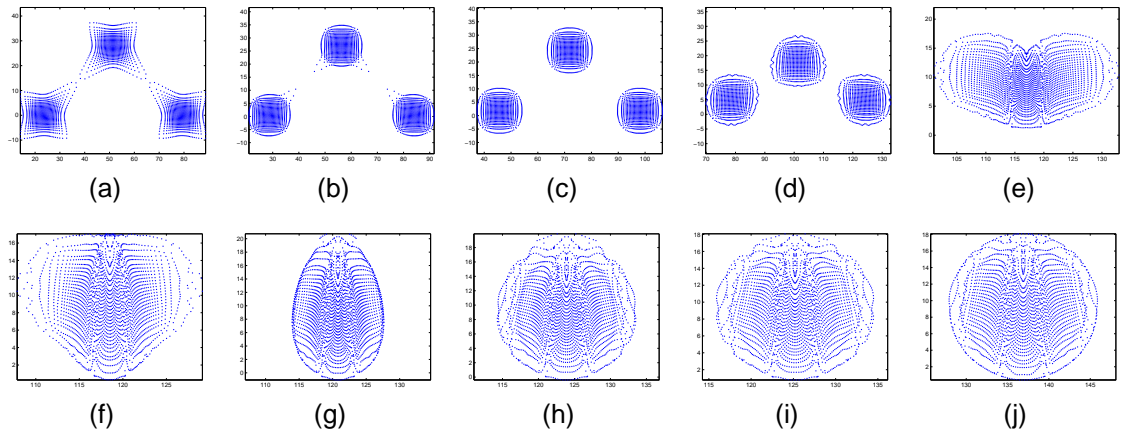


Figure 5: Swarm behavior for variable speed model at 10 different time steps,  $\tau$ : (a) 9.5, (b) 14.4, (c) 30.0, (d) 59.5, (e) 75.1, (f) 76.6, (g) 78.9, (h) 82.0, (i) 83.3, (j) 95.0.

at this point in time, the source agent (the relative position and velocities between particles) are most informative about the next state of  $p$ . Thus, as the swarm is merging into one group, the individual particles are under the most influence from their neighbors.

For  $\tau > 79.0$ , the swarm reorganized itself to find the most stable configuration, which was achieved around  $\tau = 95.0$  (Fig. 5(j)). As the swarm organized itself, both values fluctuate between local optima, while the overall values decrease. Fig. 5(h)&(i) ( $\tau = 82.0$  &  $83.3$ ) show local minimum and maximum in transfer that followed the global maximum. In these plots that the shape of the swarm fluctuates as it tries to find the most stable configuration.

It is also interesting to note that for  $85.8 < \tau < 86.2$ ,  $88.3 < \tau < 89$  and  $89.9 < \tau < 90.5$  the overall transfer entropy value dipped below 0. Negative local transfer entropy indicates that the source misleads an observer about the next state of the destination given the destination's history (Lizier et al., 2008b, 2010). This can occur when large numbers of interactions (e.g. in chaotic systems) can make the effect of any single source misleading when considered on its own. The negative transfer entropies here show that for a few time steps, most of the particles in the swarm had changes in velocity that were relatively unlikely given their relative position and velocity of their neighbors.

### Second order, constant speed model

Fig. 6 shows the total active information storage and information transfer at each time step for constant speed model. We can roughly see the three epochs of swarm dynamics in this plot:  $0 < \tau < 22.0$ ,  $22.0 < \tau < 48.0$ , and  $\tau > 48.0$ ; though the epochs are not as distinctive as those in Fig. 3.

Fig. 7 shows the average storage per particle and average transfer per causal pair at each time step for constant speed model. We can see from the storage plot in this figure that the number of interactions increases in epoch 3, since the average storage in epoch 3 is not much larger than those in

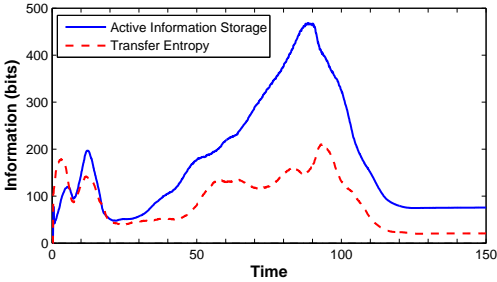


Figure 6: Total active information storage and information transfer at each time step over time for constant speed model.

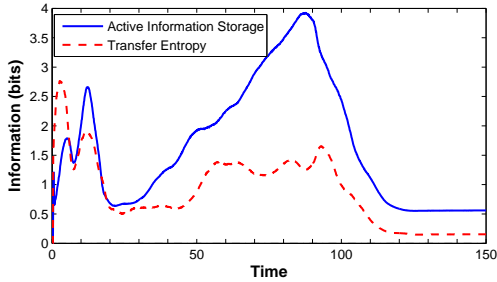


Figure 7: Average active information storage and information transfer per particle pair interaction over time for constant speed model.

epoch 1, but the total is much larger. For transfer, we can see that average value in epoch 1 is larger than those in epoch 3, which means for this model, there are more interactions with high local transfer entropy when each group of particles is “collapsing” into itself than when the groups are merging. Further, while Fig. 6 shows a definite global maximum in transfer in epoch 3, the average shows very little difference in the per interaction values. This means there are a lot more interactions at  $\tau = 93.2$  (global maximum) than at  $\tau = 58.0$  (the first local maximum in epoch 3).

Fig. 8 shows 10 snapshots of the swarm behavior at key



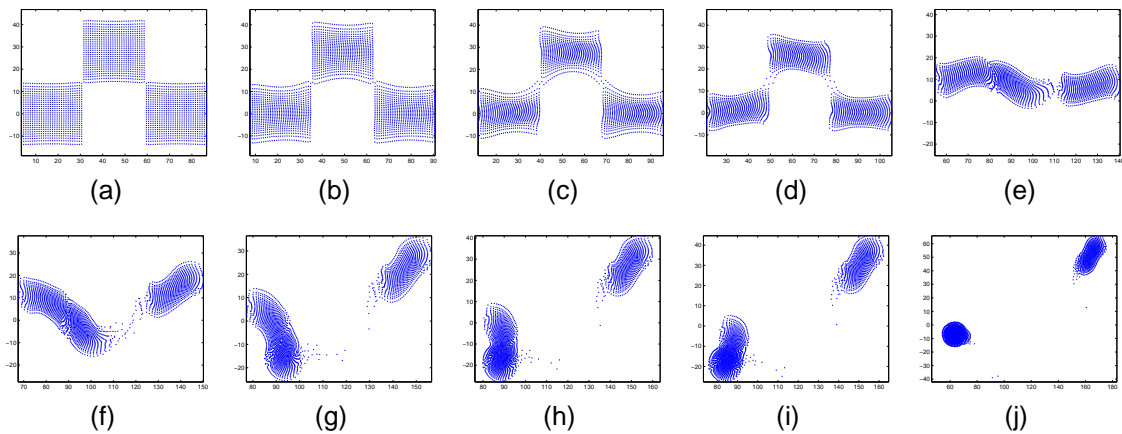


Figure 8: Swarm behavior for constant speed model at 10 different time steps,  $\tau$ : (a) 3.2, (b) 7.4, (c) 12.3, (d) 22.0, (e) 58.0, (f) 71.3, (g) 83.0, (h) 89.0, (i) 93.2, (j) 118.0.

steps as identified by Fig. 6. Fig. 8(a)-(d) shows the swarm during epoch 1, Fig. 8(e)-(i) shows the behavior during epoch 3, and Fig. 8(j) shows the final configuration when the swarm reaches steady state. These snapshots show that when particles have constant speed, the swarm organizes itself into anisotropic attractors as discussed in earlier section. Moreover, the three groups did not ultimately merge into one as in the variable speed model, but stayed as two groups.

Between  $\tau = 0$  and  $\tau = 22.0$ , both values went through a couple of oscillations before finding a steady state for the three groups. This is similar to the behavior of the swarm in variable speed model, as shown in Fig. 3. Fig. 8(a) corresponds to the first transfer local maximum, Fig. 8(b) shows the configuration when both information attain local minimum, and Fig. 8(c) corresponds to the local maximum for the storage value. While the swarm configuration does not differ much, Fig. 6 shows us that the information storage and transfer by the particles were constantly changing as the swarm organized itself.

For  $\tau > 22.0$ , as two of the groups start to merge, the transfer entropy attains the first (in epoch 3) local maximum at  $\tau = 58.0$  (Fig. 8(e)). The local minimum for transfer at  $\tau = 71.3$  is shown in Fig. 8(f), and the next local maximum at  $\tau = 83.0$  is shown in Fig. 8(g). We can see from these two snapshots that during this time the group on the left rotated  $90^\circ$  clockwise. Furthermore, we can still distinguish the two groups. The local storage increases steadily from  $\tau > 22.0$  until at  $\tau = 89.0$  (Fig. 8(h)) it reached the global maximum. At  $\tau = 93.2$  (Fig. 8(i)) the transfer entropy reached the global maximum when the group on the left finally merged into one where we cannot distinguish the boundary between the original two groups. This is similar to the plots in Fig. 3 where the global maximum for storage occurs before the global maximum for transfer.

For  $\tau > 93.2$ , both values of information dynamics decreased steadily until the swarm reaches a steady state at  $\tau = 118.0$  (Fig. 8(j)) where both values are constant.

## Discussion and Conclusions

This study examined information dynamics in swarms, applying a recently developed information-theoretic framework of distributed computation to an established individual-based swarm model. The approach verified our conjecture that swarming dynamics can be interpreted as a type of distributed computation. In particular, we adapted two localized information-theoretic measures (active information storage and transfer entropy) to the task of tracing over time how much information is stored, and how much information is transferred to and by each agent in the swarm. The state variables used in tracing the information content (stored and transferred) were chosen to be velocity and acceleration (i.e., change in velocity) — this means that the information dynamics were traced in a kinematic context. The experiments were carried out with two swarm models (variable speed and constant speed), while gathering and randomly sampling data over long transients.

The swarming dynamics were shown to be capable of exhibiting distinct configurations, including isolated groups of particles with low levels of interactions, groups that were actively merging together, and single merged groups that were retaining stable configurations, with varying degrees of inter-particle interactions. Importantly, these types, as well as transitions among these types, were shown to coincide with well-marked local and global optima of the proposed localized information-theoretic measures. Specifically, active information storage (obtained in terms of kinematics, that is, via velocity-based states) was observed to maximize during re-organization from a more fragmented to a less fragmented non-static configuration. One may argue that such a transition corresponds to an increase of kinematic order as the kinematic history begins to strongly inform an observer about the next state. Active information storage tended to minimize for either disordered/unstable configurations (with chaotic inter-particle interactions) or static con-

figurations (with low degrees of interactions): in either of these cases the kinematic history does not help the observer to predict velocity and acceleration at the next time point.

Transfer entropy, on the other hand, was observed to maximize at final stages of re-organization from more fragmented to a less fragmented non-static configuration. It may be argued that these stages correspond to the “edge of chaos” when the system actively computes its stable configuration, and the inter-particle interactions intensify as well. Transfer entropy was found to be minimal for either static or very unstable configurations (too far into the chaotic regime). Moreover, chaotic dynamics often exhibited negative local transfer entropy indicating that the source misleads the observer about the next state of the destination given the destination’s history — a sign of information modification. A detailed analysis of this aspect is left for future research.

Overall, these observations allowed us to interpret distinct phases of self-organizing swarm dynamics via the elements of distributed computation: storage, transfer, and (eventually) modification of specific kinematic information. This exemplifies once more that the process of self-organization can be described in terms of information dynamics, and makes another step towards a general theory of (guided) self-organization.

## Acknowledgements

We like to thank the High Performance Computing and Communications Centre (<http://www.hpccc.gov.au/>) for the use of their super-computer clusters in performing the experiments for this paper. LFR and JMM acknowledge the support of US National Science Foundation grants CCF-0726556 and CCF-0829748.

## References

- Aoki, I. (1982). A simulation study on the schooling mechanism in fish. *Bulletin of the Japanese Society of Scientific Fisheries*, 48:1081–1088.
- Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2003). *Self-Organization in Biological Systems*. Princeton University Press.
- Couzin, I. (2007). Collective minds. *Nature*, 445(7129):715–715.
- Couzin, I., James, R., Croft, D., and Krause, J. (2006). Social organization and information transfer in schooling fishes. In *Fish Cognition and Behavior*, Fish and Aquatic Resources, pages 166–185. Blackwell Publishing.
- Couzin, I., Krause, J., James, R., Ruxton, G., and Franks, N. (2002). Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, 218:1–11.
- Huth, A. and Wissel, C. (1992). The simulation of the movement of fish schools. *J. Theor. Biol.*, 156:365–385.
- Langton, C. G. (1990). Computation at the edge of chaos: phase transitions and emergent computation. *Physica D*, 42(1-3):12–37.
- Lissaman, P. B. S. and Shollenberger, C. A. (1970). Formation flight of birds. *Science*, 168(3934):1003–1005.
- Lizier, J. T., Prokopenko, M., and Zomaya, A. Y. (2007). Detecting non-trivial computation in complex dynamics. In *Proc. of the 9th European Conf. on Artificial Life (ECAL), Lisbon, Portugal*, volume 4648 of *Lecture Notes in Artificial Intelligence*, pages 895–904, Berlin / Heidelberg. Springer.
- Lizier, J. T., Prokopenko, M., and Zomaya, A. Y. (2008a). The information dynamics of phase transitions in random Boolean networks. In *Proc. of the Eleventh Int. Conf. on the Simulation and Synthesis of Living Systems (ALife XI), Winchester, UK*, pages 374–381, Cambridge, MA. MIT Press.
- Lizier, J. T., Prokopenko, M., and Zomaya, A. Y. (2008b). Local information transfer as a spatiotemporal filter for complex systems. *Physical Review E*, 77(2):026110.
- Lizier, J. T., Prokopenko, M., and Zomaya, A. Y. (2010). Information modification and particle collision in distributed computation. *Chaos*, 20(3):037109.
- Lizier, J. T., Prokopenko, M., and Zomaya, A. Y. (2011). Local measures of information storage in complex distributed computation. Under submission.
- Lukeman, R., Li, Y.-X., and Edelstein-Keshet, L. (2010). Inferring individual rules from collective behavior. *Proceedings of the National Academy of Sciences of the United States of America*, 107(28):12576–12580.
- MacKay, D. J. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, Cambridge.
- Miller, J. M., Kolpas, A., Juchem-Neto, J. P., and Rossi, L. F. (2011). A continuum three-zone model for swarms. *Bull. Math. Bio.*, To appear.
- Miramontes, O. (1995). Order-disorder transitions in the behavior of ant societies. *Complexity*, 1(3):56–60.
- Parrish, J. K. and Edelstein-Keshet, L. (1999). Complexity, pattern, and evolutionary trade-offs in animal aggregation. *Science*, 284(5411):99–101.
- Parrish, J. K., Viscido, S. V., and Grünbaum, D. (2002). Self-organized fish schools: An examination of emergent properties. *The Biological Bulletin*, 202:296–305.
- Partridge, B. L. (1982). The structure and function of fish schools. *Scientific American*, 246(6):114–123.
- Schreiber, T. (2000). Measuring information transfer. *Physical Review Letters*, 85(2):461–464.
- Sinclair, A. R. E. and Norton-Griffiths, M., editors (1979). *Serengeti: Dynamics of an Ecosystem*. University Of Chicago Press.
- Uvarov, B. P. (1928). *Grasshoppers and Locusts*. Imperial Bureau of Entomology.
- Vicsek, T., Czirok, A., Ben-Jacob, E., Cohen, I., and Shochet, O. (1995). Novel Type of Phase Transition in a System of Self-Driven Particles. *Phys. Rev. Lett.*, 75:1226–1229.

# An agent-based model of jaguar movement through conservation corridors

Angela Watkins<sup>1</sup>, Jason Noble<sup>1</sup> and C. Patrick Doncaster<sup>1</sup>

<sup>1</sup>Institute for Complex Systems Simulation, University of Southampton  
aw4g09@soton.ac.uk

## Abstract

Wildlife corridors mitigate against habitat fragmentation by connecting otherwise isolated regions, bringing well-established benefits to conservation both in principle and practice. Populations of large mammals in particular may depend on habitat connectivity, yet conservation managers struggle to optimise corridor designs with the rudimentary information generally available on movement behaviours. We present an agent-based model of jaguars (*Panthera onca*), scaled for fragmented habitat in Belize where proposals already exist for creating a jaguar corridor. We use a least-cost approach to simulate movement paths through alternative possible landscapes. Six different types of corridor and three control conditions differ substantially in their effectiveness at mixing agents across the environment despite relatively little difference in individual welfare. Our best estimates of jaguar movement behaviours suggest that a set of five narrow corridors may out-perform one wide corridor of the same overall area. We discuss the utility of ALife modelling for conservation management.

## Introduction

One of the most obvious effects of our own species on the planet has been the clearing of forests to make way for agriculture. In many parts of the world this means that the natural vegetation that remains tends to be divided into isolated patches (see figure 2 for an illustration) with disruptive consequences for the local wildlife. The establishment and maintenance of “corridors” connecting otherwise isolated areas of habitat have therefore been put forward as important tools in conservation biology (Bennett, 2000; Hilty et al., 2006). The idea of a corridor is to connect local sub-populations into a single meta-population and thereby reduce the risk of local extinctions due to human activity (hunting, land development, etc.) and, more importantly, to improve the species’ long-term survival chances by increasing the size of the gene pool.

Bennett (2000) shows that evidence for the effectiveness of habitat corridors is mixed: they have been more helpful for some species than others. Indeed, habitat fragmentation is itself a concept that depends on the details of the behavioural ecology of the species concerned (consider, for



Figure 1: A jaguar photographed using a stealth camera. Image courtesy of the Jaguar Corridor Initiative, Belize.

example, the difference between a bird and a snail in their ability to move between habitat patches). The current paper puts forward a simulation model to help assess the effectiveness of different corridor policies for the jaguar, *Panthera onca*.

The jaguar (figure 1) is an apex predator that stalks and ambushes its prey. It is the third-largest of the big cats and the largest big cat species in the Western hemisphere. Its range extends from the southern United States to northern Argentina. Jaguars are stealthy and elusive, and thus there is still much we do not know about their behaviour. However, one of the better-studied jaguar populations is in Belize, on the Caribbean coast of Central America. In particular, the Cockscomb Basin Wildlife Sanctuary (CBWS), a 425 square-km reserve in southern Belize, has been a productive jaguar fieldwork site for several decades (Rabinowitz and Nottingham, 1986a; Harmsen et al., 2010b). Biologists working there have been instrumental in setting up the Jaguar Corridor Initiative (Rabinowitz and Zeller, 2010), a cooperative effort between scientists, conservation groups, and regional governments to establish corridors connecting known jaguar populations.

Assessing the usefulness of a corridor initiative is diffi-

cult when we do not fully understand the behaviour of the species involved. Two of us (AW and CPD) are conducting ongoing fieldwork at the CBWS in Belize, but we recognize that data on jaguar numbers and movement, collected through means such as stealth cameras and radio-tracking, will not be sufficient on its own. Such data collection efforts need to be combined with modelling in order to improve our understanding of jaguar behaviour. There has been some recent progress on statistical, data-driven modelling in this regard (see for example the Bayesian approach of Colchero et al., 2011) but we believe there is also utility in the agent-based modelling approach characteristic of work in artificial life.

Agent-based models explicitly represent the behaviours of individual organisms, allowing us to simulate both the interactions between individuals, and those between the individual and the environment (Grimm, 1999). For our purposes, the advantages of these types of models are the ability to integrate individual behaviours with landscape dynamics, to model individual-level adaptive processes such as learning and memory, and to study collective responses to changes in landscape composition. The potential to explore many alternative scenarios also provides distinct advantages over classical ecological models.

Agent-based modelling approaches have been widely used already, of course, under the banners of both artificial life and of ecology, to study the movement of animals through their environments. Examples include Nonaka and Holme's (2007) model of optimal foraging in clumpy environments, Wheeler and de Bourcier's (2010) work on the evolution of territorial signalling, and Hemelrijk's (1998) model of the spatial aspects of dominance hierarchies in chimpanzees.

In constructing a model of jaguars moving around in their habitat and using (or *not* using) corridors, we will need a way to model their decision-making about where to go next. This is an opportunity to integrate the "least-cost modelling" paradigm from landscape ecology (Adriaensen et al., 2003) with the agent-based approach. The idea behind least-cost modelling is simple: it is a species-specific calculation based on the assumption that dispersing organisms are more likely to use a route of least resistance when traversing a landscape. In other words, whenever they are faced with a choice while moving around their spatial network, they will choose the lowest-cost option. Cost estimates are themselves derived from data on how frequently the animals are observed in particular landscape types, and their preference for one type over another in choice tests.

Least-cost modelling techniques are standard in many GIS (Geographical Information System) packages which offer built-in cost and distance functions that allow for rapid model construction (Rayfield et al., 2010). A raster-based grid of the landscape is generated with a cost assigned to each cell that represents the lowest cumulative cost from

that cell to the source cell. This cost is the inverse of the degree of functional connectivity of the landscape according to the species in question (Driezen et al., 2007) and thus the end product of the calculation can be seen as a probability distribution across the landscape describing the likelihood of the animal settling at any given position. Rabinowitz and Zeller (2010) developed an ambitious least-cost model of jaguar dispersal across their entire range in Central and South America.

Validating least-cost models is not easy, however. Driezen et al. (2007) produced one of the only studies to successfully compare the output of least-cost models with empirical data on animal movement. They used statistics on landscape-wide cost values and compared these to real hedgehog paths, constructing and presenting a novel approach to matching empirical movement trajectories with generated least-cost maps. Watkins (2010) demonstrated that this approach could be taken further through integration with agent-based modelling.

The aim of the current project is to build a simple agent-based model of jaguar behaviour, employing a least-cost view of movement, in order to look at how the spatial structure of corridors intended to connect disjoint forest habitats could affect conservation goals. In short, we ask the reader to imagine two separated expanses of forest (as occurs in many locations in Belize) and enough resources to protect a few tens of square kilometres of remnant forest from further disturbance and human development. What would be the best corridor design policy? One wide corridor? Multiple thin corridors? A series of small "islands" between the two forests? How much could we expect of such a corridor once constructed, i.e., what effects would it have on individual welfare and on genetic mixing at the population level? We contend that the answers to these questions will be an emergent function of jaguars' preferences for different landscape types and their territorial interactions with each other.

This work is intended to be the first in a series of increasingly detailed models of jaguar ecology. The integration of real GIS data into the model is beyond the scope of the current study — we think there are basic questions to ask of an abstract model first — but is the logical next step for future models. Basing simulated models in real landscapes can only improve our ability to draw conclusions about system-level behaviours in realistic environments.

## The model

The first step in constructing our model is devising a map layout that reflects the essentials of the problem. Figure 2 shows a typical Belizean landscape and illustrates the fragmentation of forest habitat that occurs due to road construction, tree-clearing for farming, urban development, etc. The key feature of our simulation will thus be two separated blocks of forest, surrounded by cleared farmland. Each forest section will hold an initial population of jaguars; the



Figure 2: An aerial view of a typical landscape in Belize. Note that regions of ideal jaguar habitat (i.e., forest) are separated by roads and cleared farmland. Image: Google Earth.

question is how easy or difficult it will be for them to travel from one forest zone to another.

Figure 3 shows the potential corridor designs that we will investigate. We begin with the basic two-forest layout in the top left corner. Note the blue edges where the forest meets farmland; we assume that these transitional zones are of intermediate appeal to the jaguars. The next design (top centre) features a corridor connecting the two forest sections. We also consider (top right) a layout with additional area added to the forest sections: this is equivalent to a control condition in which we spend the conservation budget on extending each forest rather than connecting them. Next we consider whether corridor width is more or less important than the number of corridors by looking at three- and five-corridor designs. In each case the same total area is devoted to the connecting corridors. These are followed by one- and three-island designs — alternatives to a direct corridor — and a design made up of many randomly placed islands. Again, the total area devoted to corridor is a constant. Finally we also look at a “contiguous forest” layout where the entire map is forested: this is another control condition in that it allows us to compare jaguar ecology in a modern fragmented habitat with what it might have been before human colonization.

The map is not meant to be a precise rendition of any particular location, but we do need to establish a scale in order to incorporate what is known about jaguar population density, movement rates, and territory size (our primary references in this were Schaller and Crawshaw, 1980; Harmsen et al., 2010c). The map is represented as a  $100 \times 100$  grid of squares, with each square being 500 metres on a side. This means that the entire map covers  $50 \times 50$  km, with each of

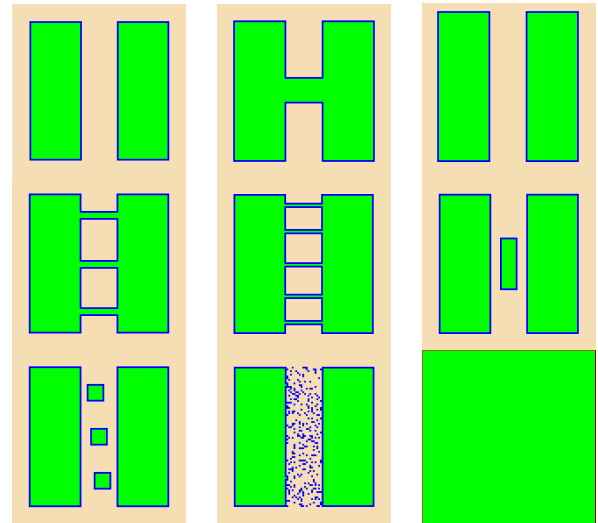


Figure 3: Map layouts investigated in the simulation. Core forest is in green, forest edges are blue, and farmland is khaki. First row: no corridor, one corridor, no corridor but equivalent area added to the forest. Second row: three corridors, five corridors, one island. Third row: three islands, random islands, contiguous forest.

the basic forest sections measuring  $15 \times 40$  km and with a 10 km expanse of farmland between them. For comparison, the 2500 square km area of the map represents about 10% of the land area of Belize.

In most layouts the map includes 1275 square km of forest (the exceptions are the no corridor layout with 1200 square km and the contiguous forest condition with 2500 square km). Each run of the simulation begins by placing 100 jaguars into randomly chosen forest squares, which corresponds to a density of 7.84 jaguars per 100 square km. This is consistent with Rabinowitz and Nottingham (1986b) who found a minimum home range size of 10 square km per animal, and also with Harmsen et al. (2010c) who estimated densities of 3.5 to 11.0 individuals per 100 square km in the CBWS, which is itself thought to be a “hot spot” for jaguar numbers. Our simulated population of 100 jaguars thus represents a medium to high population density.

Edge effects are known to be important in landscape ecology, and so we added an edge-detecting routine to the initialization of our map. Any forest square that borders a farmland square (in any of 8 neighbouring positions) is labelled as an edge square. These are shown in blue in figure 3.

What about temporal scale? Schaller and Crawshaw (1980) recorded daily travel of between 1 km and 3 km straight-line distance for jaguars, with males travelling further than females. In our model male jaguars move one grid square every timestep; if all eight surrounding squares have equal cost, the movement will be in a random direction. In order to get plausible straight-line daily travel distances





know the exact values that should be plugged in. The point is not to make a precise predictive model but to see whether it is possible to explain the basics of jaguar movement with some simple rules. In this regard, we do have circumstantial evidence: jaguars have occasionally been observed in pastures both in Belize and Brazil, and we know that jaguars are somewhat territorial. If we chose much higher values for the landscape cost of farmland, the jaguars would not leave the forest at all, even under extremely crowded local conditions. Conversely, if we make the cost of encountering another jaguar's pheromone too high, the animals will spill out into the farmland in great numbers in an effort not to encroach on each other's territory.

Our simple pheromone mechanism is actually a reasonable model of how jaguars maintain their territorial boundaries in the real world. Jaguars are not as likely to mark their passage with urine or scat as other felids are (Schaller and Crawshaw, 1980; Harmsen et al., 2010c) but they are known to scent-mark by scraping trees in their territory (Harmsen et al., 2010a).

There is one more cost to be considered: we also made the jaguars sensitive not just to pheromones deposited by others but also to their *own* pheromone trails. The cost of entering a grid square where you were the last occupant is equal to 15% of the pheromone level (i.e., the effect is about 7 times weaker than for the pheromones of others). This reflects the fact that a section of forest where the animal has not hunted recently is a better prospect for prey than the same grid-square they occupied the day before. The effect is to stop the jaguars back-tracking on their own path. A solitary jaguar in a large expanse of forest will therefore perform a random walk strongly biased towards yet-unvisited grid squares, in effect carving out a territory of maximal size for itself.

Unlike much ALife work, there is no genetic algorithm in our model: our central question is not evolutionary but ecological. In the same vein as Hemelrijk (1998) we are not asking about the evolution of the animals' strategies, but about the implications of how a hypothesized behavioural program would play out when followed by multiple animals in a simulated spatial world.

The goal is to use our model of jaguar movement behaviour to evaluate the effectiveness of different corridor layouts — but what can we measure in order to do that? The jaguars' behavioural strategies are not evolving, so we cannot measure “fitness” per se. Instead we look at the average cost level for the grid squares each jaguar chooses to enter over the course of the run. This is effectively a measure of “jaguar welfare”. Low cost grid squares (i.e., what jaguars want) are places in the forest that have not recently been visited by other jaguars. The low cost ultimately reflects the fitness benefits of being in such places: these are areas with high prey availability, low risk of being killed by farmers, low risk of costly fights with other jaguars, etc. Higher val-

ues on the average-cost measure will therefore be associated with stress or over-crowding. If one corridor layout can reduce this value compared to another, this is evidence for its jaguar-conservation effectiveness.

We are not simulating enough detail of the jaguar's lifestyle to look at mating behaviour directly, but we can look indirectly at whether different corridor layouts would encourage a larger breeding population as opposed to isolated sub-populations. We have done this simply by recording the proportion of jaguars that finish the year on the opposite side (east-west) of the map compared to where they started. A value of 0% indicates two isolated sub-populations, whereas 50% would indicate random mixing.

## Results

Figure 4 shows a typical screenshot from the simulation. We can see that the model has been successful in reproducing male territories of a plausible size of 10 to 20 square km, and that a minority of jaguars have resorted to hunting in farmland. When watching the animation over time it is very easy to interpret the jaguar movements as “patrolling” a territory and avoiding conflicts with each other; the forest edges are used as “pathways” around territories; established core territories shift only gradually; and the jaguars that are forced out into farmland eventually get back into the forest when they are lucky enough to find an undefended edge section. Figure 4 shows the “one corridor” layout, and we can see that the corridor is certainly occupied by jaguars and thus might be leading to genetic mixing between the two sub-populations.

However, we can also see a threat to this exchange: note that the brown and the yellow territories in the centre of the corridor act as barriers to the transit of any other (male) jaguars. Our qualitative impressions when watching the simulation run with different corridor layouts were that the geography of the corridor could certainly make a difference as some layouts, notably the five-corridor map, led to “channeled” movement back and forth across the corridor, whereas other layouts such as the one in figure 4 led to blockages.

Figure 5 shows the comparison of the average-cost values across all 9 conditions. The obvious pattern here was that the layout did not seem to make a great deal of difference to the average cost experienced by each animal, except in the “contiguous forest” case. It is obvious that the contiguous layout will lead to lower average costs, however, as the same number of jaguars are distributed across about twice as much forest, giving larger territory sizes and fewer encounters with the pheromones of others.

The “no corridor” and “random islands” conditions lead to slightly higher costs than in other conditions. In the former case this is simply because there is less forest territory available; in the “equal area” control condition this difference disappears. The “random islands” condition leads to



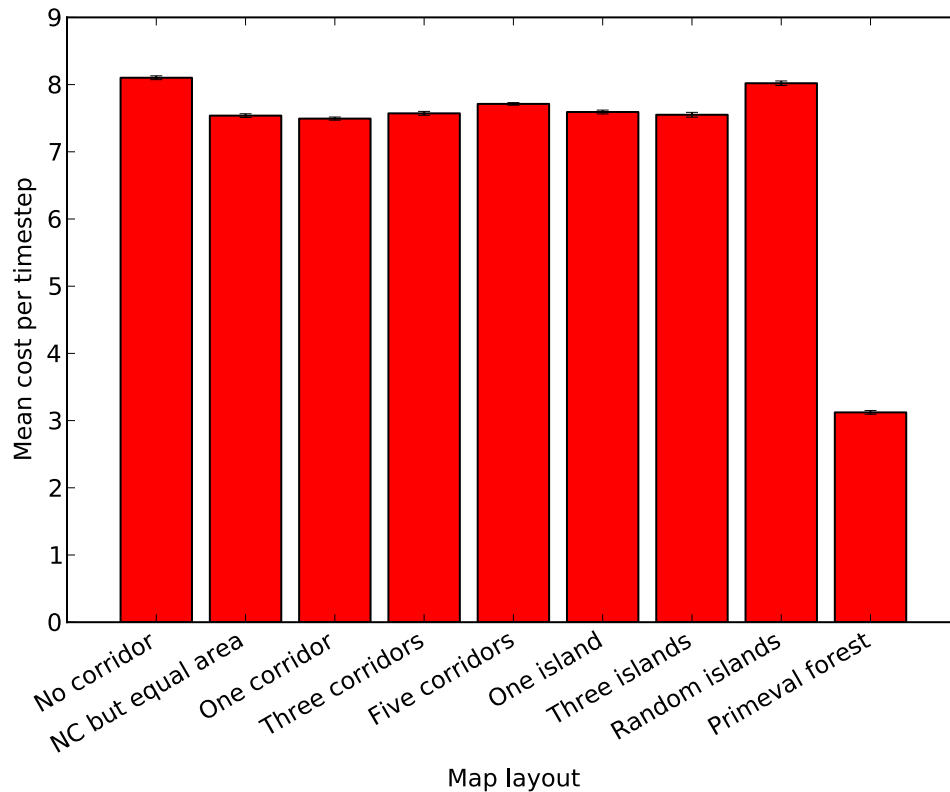


Figure 5: Mean cost figures per jaguar per timestep compared across the nine different map layouts. Standard errors are calculated across 25 replications of each condition with different random seed values.

most of the corridor squares being edge squares, and there is a concomitant increase in average cost. On this evidence it would seem that corridor design does not make much difference to jaguar welfare, and that the critical thing is simply to have as much favourable habitat available as possible.

What of the genetic mixing results? If we look at figure 6 we see the mean level of movement across the centre-line of the map, over the different conditions. The differences here are much more dramatic. The “contiguous forest” condition is again the most favourable for the jaguars, with 34% mixing (approaching the 50% level that you would get if the jaguar locations were shuffled at random). This contrasts with the “no corridor” conditions that support only 7 or 8% mixing. The island-based corridor designs perform very badly as well, although things are not quite so bad with the “random island” design. The striking finding from figure 6 is that corridor-based designs perform best, and that the more corridors and/or the thinner the corridor, the better. Observation of these runs suggests that the strong performance of the five-corridor design (26% swapping) is because the thin pathways promote rapid movement, often through the edge squares if another animal has recently

passed through the forest squares, and the very thin strip of core forest (just 500 metres wide) is not big enough to support a territory. Wider corridors (the three-corridor and the one-corridor cases) were better than island-based designs, and certainly better than no corridor at all, but did not match the mixing levels of the five-corridor case due to the tendency for the corridor to become blocked by an established territory.

## Conclusions

We were pleased with the qualitative results of the model in that we managed to replicate plausible territorial behaviour in jaguars using the least-cost paradigm and only a few assumptions. The model has brought novel aspects of the corridor design problem to light, notably the possibility that some corridor layouts could be counter-productive due to being large enough to support internal territories that then acted as obstacles to travel by other animals. We feel that the agent-based modelling approach we have begun here has the potential to be extremely useful in drawing out the implications for different theories about jaguar behaviour and thereby helping to determine which of those theories is a

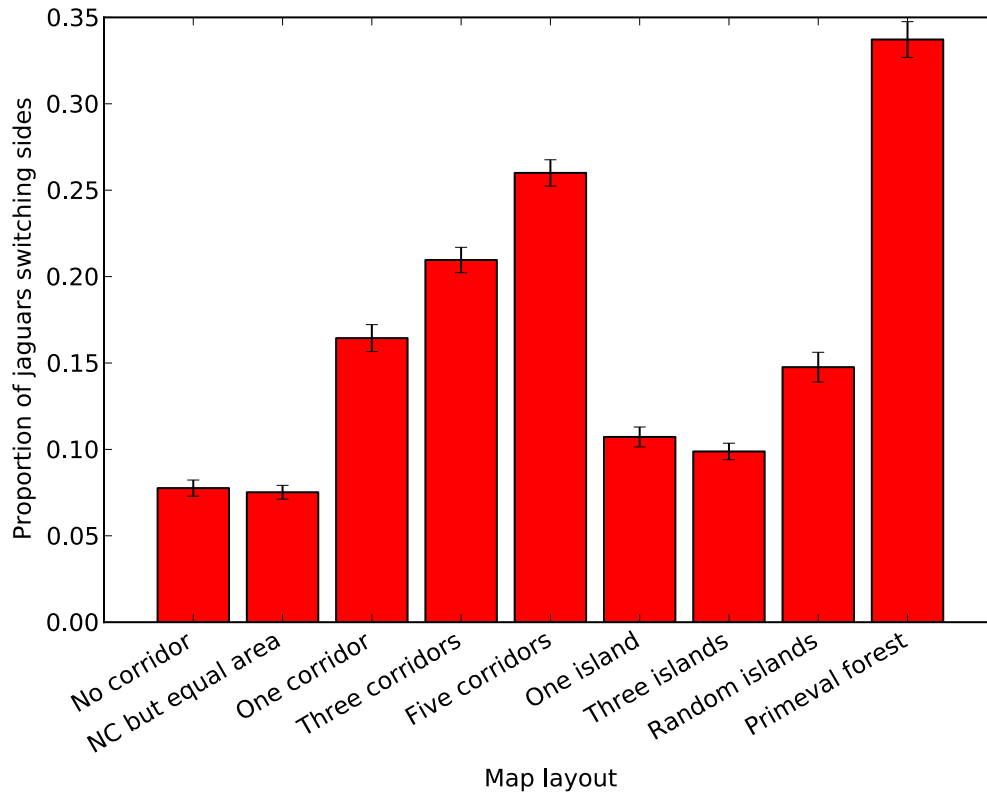


Figure 6: Proportion of jaguars that move from one side of the map's centre-line to the other by the end of the simulated year, compared across the nine different map layouts. Standard errors are calculated across 25 replications of each condition with different random seed values.

better match for the multi-faceted and incomplete observational data we have on the real animals (see Di Paolo et al., 2000, for an account of how this process can work). There are many parameters in the model for which we have had to guess at an appropriate value, but the idea is to take these values as a starting point and use them in an iterative process of model refinement in future comparisons with empirical data from Belize.

We began our modelling with a hypothetical question about the best corridor design to choose if you had the resources to reforest a few tens of square km of Belizean farmland separating two forests. We can answer that question unequivocally: of the corridor layouts we explored, the five-corridor layout was the most effective. We had expected that we might see significant differences in the average landscape cost value experienced by the jaguars across the different corridor designs, but this turned out not to be the case. Average landscape cost, given a constant population of jaguars, seems to be explained almost entirely by the availability of core forest grid squares. This suggests, for example, that constructing a new conservation corridor in Belize would

not lead to a big boost in the landscape's carrying capacity for jaguars. Instead, the key difference observed between our corridor designs was their capacity to promote migration from one side of the map to the other, and thus to promote genetic mixing at the whole-population level. The five-corridor case achieved levels of cross-map migration that were almost comparable to the "contiguous forest" condition, which is a great outcome from a conservation perspective.

Having established that this agent-based least-cost modelling approach is viable, there are several ways in which we could improve the model. Incorporating real maps of the Belizean landscape using GIS packages is an obvious way of increasing the model's fidelity, although we believe it is important not to rush this process: we need to understand the dynamics of how our simulated jaguars behave in simplified environments first. Still, using GIS data would also allow us to build a richer least-cost model, incorporating data on jaguar preferences for entering or avoiding terrain such as hills, differing densities of forest, roads, and urban areas.

In terms of the corridor design problem, a weakness of

the current model is that we only compared six specific corridor layouts with three control conditions. If we settled on a way to represent the spatial layout of a corridor, e.g., as a bitmap, we could use a genetic algorithm or other optimization technique to search for the best *possible* layout for the connecting corridors. This is perhaps slightly premature at this stage as the model is in an exploratory mode; we do not yet know enough about jaguar movement behaviour to be sure that such an optimized layout would be accurate enough to serve as a reliable conservation policy recommendation. Nevertheless we would at least be in a position to say *why* we believed a certain corridor design was optimal.

In conclusion: jaguars are rare, elusive, and hard to study. In coming years, we expect that improvements in radio- and GPS-tracking technology should see an increase in the data we have available on how they move around their environment. However, as that data comes in, it will be important to be able to evaluate it in the light of competing theories about how jaguars make decisions about hunting, mating, territory defence, etc. The agent-based simulations of artificial life can clearly help in doing this.

## Acknowledgements

AW was supported by the UK's Engineering and Physical Sciences Research Council via Southampton's Institute for Complex Systems Simulation and doctoral training centre. CPD was supported by Darwin Initiative grant 17-012 from the UK Department for Environment, Food and Rural Affairs. We also wish to acknowledge the assistance of the Jaguar Corridor Initiative.

## References

- Adriaensen, F., Chardon, J., De Blust, G., Swinnen, E., Villalba, S., Gulinck, H., and Matthysen, E. (2003). The application of 'least-cost' modelling as a functional landscape model. *Landscape and Urban Planning*, 64:233–247.
- Bennett, A. F. (2000). *Linkages in the Landscape: The Role of Corridors and Connectivity in Wildlife Conservation*. International Union for Conservation of Nature, Gland, Switzerland.
- Colchero, F., Conde, D. A., Manterola, C., Chávez, C., Rivera, A., and Ceballos, G. (2011). Jaguars on the move: modeling movement to mitigate fragmentation from road expansion in the Mayan Forest. *Animal Conservation*, 14(2):158–166.
- Di Paolo, E., Noble, J., and Bullock, S. (2000). Simulation models as opaque though experiments. In Bedau, M., McCaskill, J., Packard, N., and Rasmussen, S., editors, *Artificial Life VII: Proceedings of the Seventh International Conference on Artificial Life*, pages 497–506. MIT Press, Cambridge, MA.
- Driezen, K., Adriaensen, F., Rondinini, C., Doncaster, C., and Matthysen, E. (2007). Evaluating least-cost model predictions with empirical dispersal data: A case-study using radio-tracking data of hedgehogs (*erinaceus europaeus*). *Ecological Modelling*, 209:314–322.
- Grimm, V. (1999). 10 years of individual-based modelling in ecology: what we have learned and what we could learn in the future. *Ecological Modelling*, 115:129–148.
- Harmsen, B. J., Foster, R. J., Gutierrez, S. M., Marin, S. Y., and Doncaster, C. (2010a). Scrape-marking behavior of jaguars (*Panthera onca*) and pumas (*Puma concolor*). *Journal of Mammalogy*, 91:1225–1234.
- Harmsen, B. J., Foster, R. J., Silver, S. C., Ostro, L. E. T., and Doncaster, C. P. (2010b). The ecology of jaguars in the Cockscomb Basin Wildlife Sanctuary, Belize. In Macdonald, D. W. and Loveridge, A., editors, *The Biology and Conservation of Wild Felids*, pages 403–416. Oxford University Press.
- Harmsen, B. J., Foster, R. J., Silver, S. C., Ostro, L. E. T., and Doncaster, C. P. (2010c). The ecology of jaguars in the Cockscomb Basin Wildlife Sanctuary, Belize. In Macdonald, D. W. and Loveridge, A., editors, *The Biology and Conservation of Wild Felids*, pages 403–416. Oxford University Press.
- Hemelrijk, C. (1998). Spatial centrality of dominants without positional preference. In Adami, C., Belew, R., Kitano, H., and Taylor, C., editors, *Artificial Life VI*, pages 307–315. MIT Press, Cambridge, MA.
- Hilty, J. A., Lidicker, W. Z., and Merenlender, A. M. (2006). *Corridor Ecology: The Science and Practice of Linking Landscapes for Biodiversity Conservation*. Island Press, Washington, DC.
- Nakamura, M. and Kurumatani, K. (2008). Formation mechanism of pheromone pattern and control of foraging behaviour in an ant colony model. In Langton, C. and Shimhara, K., editors, *Artificial Life V*, pages 67–74. MIT Press, Cambridge, MA.
- Nonaka, E. and Holme, P. (2007). Agent-based model approach to optimal foraging in heterogeneous landscapes: effects of patch clumpiness. *Ecography*, 30:777–788.
- Rabinowitz, A. and Nottingham, B. (1986a). Ecology and behaviour of the jaguar (*Panthera onca*) in Belize, central America. *Journal of Zoology*, 210(1):149–156.
- Rabinowitz, A. and Zeller, K. (2010). A range-wide model of landscape connectivity and conservation for the jaguar, *Panthera onca*. *Biological Conservation*, 143:939–945.
- Rabinowitz, A. R. and Nottingham, Jr., B. G. (1986b). Ecology and behaviour of the jaguar (*Panthera onca*) in Belize, Central America. *Journal of Zoology*, 210(1):149–159.
- Rayfield, B., Fortin, M., and Fall, A. (2010). The sensitivity of least-cost habitat graphs to relative cost surface values. *Landscape and Ecological Engineering*, 25:519–532.
- Schaller, G. B. and Crawshaw, Jr., P. G. (1980). Movement patterns of jaguar. *Biotropica*, 12(3):161–168.
- Watkins, A. (2010). Modelling mammalian movements: exploring and integrating least-cost models and agent-based simulations in fragmented landscapes. Master's thesis, University of Southampton.
- Wheeler, M. and de Bourcier, P. (2010). How not to murder your neighbour: using synthetic behavioural ecology to study aggressive signalling. *Adaptive Behaviour*, 3(3):273–309.

# Transformations and Multi-Scale Optimisation in Biological Adaptive Networks

Richard A. Watson<sup>1</sup>, Rob Mills<sup>1</sup> and C. L. Buckley<sup>2</sup>

<sup>1</sup>Natural Systems group, Electronics and Computer Science, University of Southampton, U.K.

<sup>2</sup>Informatics, Sussex University, U.K.

raw@ecs.soton.ac.uk

The natural energy minimisation behaviour of a dynamical system can be interpreted as a simple optimisation process, finding a locally optimal resolution of constraints between system variables. In human problem solving, high-dimensional problems are often made much easier by inferring a low-dimensional model of the system in which search is more effective. But this is an approach that seems to require top-down domain knowledge; not one amenable to the spontaneous energy minimisation behaviour of a natural dynamical system. However, in recent work we investigated the ability of distributed dynamical systems to improve their constraint resolution ability over time by self-organisation. Using a ‘self-modelling’ Hopfield network with a particular type of associative connection we illustrated how slowly changing relationships between system components results in a transformation into a new system, a low-dimensional caricature of the original system, in which the energy minimisation behaviour is significantly more effective at globally resolving system constraints. This uses only very simple and fully-distributed positive feedback mechanisms that are relevant to other ‘active linking’ and adaptive networks. Here we overview the implications of this neural network model for understanding transformations and emergent collective behaviour in various non-neural adaptive networks such as social, genetic and in particular, ecological networks.

**Optimisation in Dynamical Systems.** Physical dynamical systems with a large number of simple equivalent components have been shown to exhibit ‘emergent collective computational abilities’ [8] such as implementing content-addressable memory or solving constraint satisfaction problems [9,10]. In the latter, Hopfield and Tank equate the energy minimisation behaviour of a dynamical system with an optimisation process – i.e., the system moves to configurations that better-resolve the conflicting constraints between system variables. But actually, energy minimisation in a simple dynamical system is equivalent to the simplest possible optimisation algorithm, namely gradient descent (or incremental improvement), which in anything but the simplest of problems tends to find only locally optimal solutions. In human design-engineering and optimisation, high-dimensional problems are often made much easier by inferring a low-dimensional model of the system (e.g., a high-level representation that exploits modularity/problem decomposition), such that local search in this new space is better able to find a globally optimal resolution of constraints. This is an approach that seems to require top-down domain knowledge and design intelligence, but sophisticated model-building algorithms can exploit such an approach bottom-up by learning and exploiting problem structure from observed

correlations, probing epistatic interactions, or simply ‘memoising’ hard-won partial solutions [18,16,15,14,19,2]. Nonetheless, such approaches do not appear to be amenable to the spontaneous energy minimisation behaviour of a simple dynamical system. But can other types of dynamical systems, specifically self-organising systems, perform more sophisticated forms of optimisation? And conversely, can an optimisation framework help us to better understand the behaviour of natural self-organising systems?

Our questions are motivated by consideration of self-organising multi-agent systems, such as species in an ecosystem or agents in a socio-economic network, and their potential to exhibit emergent collective behaviours. In particular, we are interested in the possibility that such systems can spontaneously transform into a new system, operating at a higher level of organisation [21,13], and that such a dynamical transformation may facilitate (or may even be equivalent to) a transition in the ability to resolve constraints between the system components. Abstractly, these systems can be characterised as ‘adaptive networks’ [5] sharing the property that the structure of connections between agents affects changes to the agent behaviours and, vice versa, that the agent behaviours affect changes to the structure of connections between agents. The Hopfield network [8] easily accommodates such state/topology coadaptation and, at a very abstract level, provides a suitable system to explore how self-organisation in adaptive networks alters their ability to resolve conflicting constraints between system components.

**Transformations in meta-dynamical systems/adaptive networks.** In formalising the behaviour of a system that transforms its dynamics over time we cannot treat the parameters of the system as fixed – instead we need to pull them into the model such that they become variables controlled from within the model. But we characterise a transforming system as a ‘meta-dynamical system’ [3]. That is, the network topology defines the parameters of the state dynamics, but the connections of this topology are in actuality (slow changing) variables. In the sub-space of state dynamics defined for any given topology, or in the larger joint space of state variables and topology together, we simply observe a dynamical system doing what it does naturally, minimizing energy – there is no sense in which the system is ‘improving’ its ability to minimise energy. But when we regard the connections as ‘changing parameters’ of the state dynamics, then we can characterise these changes in terms of how they transform the dynamics of the state variables. In particular, we

can assess whether this transformation *improves* the ability of the state dynamics to minimise energy.

Using a ‘self-modelling’ [23] Hopfield network with Hebbian learning [1,6] as a model adaptive network, we recently showed that it is possible for simple distributed mechanisms, gradually changing the connections of the network, to cause it to effectively rescale its dynamics and hence move from local to global energy minimization by encapsulating implicit dynamical sub-structure [26] (Fig.1).

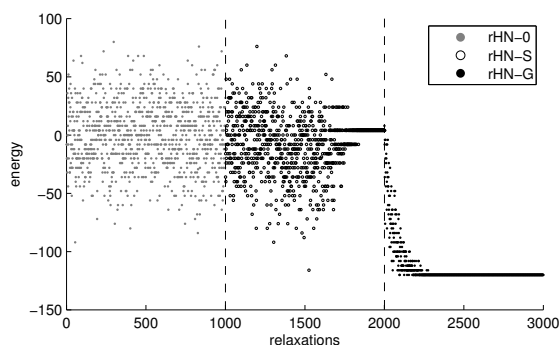


Fig.1: The distribution of attractor-state energies found over time in a restart Hopfield network without learning (rHN-0), with ordinary Hebbian learning (rHN-S), and with ‘generative associations’ (rHN-G). The latter transforms the system into one which easily and reliably minimises total energy. See [26].

## Transformations in Biological Adaptive Networks.

Although this recent work utilised a Hopfield network with Hebbian learning, a separate recent result shows that the same type of behaviour is expected spontaneously in other (non-neural) adaptive networks [25]. Specifically, when individual self-interested agents on a network can alter network connections (e.g. alter their fitness dependencies with others by changing their resource-utilisation profile, or alter the proportion of time/resources they invest in a relationship, or alter the probability of interaction or co-dispersal with others) and they do so to maximise their individual utility then the alterations they choose are necessarily Hebbian. Intuitively, this occurs because short-sighted selfish agents reinforce the status-quo [4], or increase the robustness/stability of the current state configuration [24], and this has the same dynamical consequences on the subsequent dynamics of the system as Hebbian learning does when it stores a training pattern in a neural network. Accordingly, related work develops the implications of this model for genetic networks [24] (with relation to evolvability and robustness [11,17]), social networks [4] (games on networks with active linking [20]) and ecological networks [12] and finds that the same dynamics occur spontaneously in all these systems. In this presentation we focus particular attention on ecosystems and the ‘generative’ type of associations (Fig.1) that have the effect of forming coalitions [20,28] or new selective units [27,22]. We suggest that this provides a formal framework for characterising the selective pressures/adaptive consequences involved in the formation of evolutionary transitions [13].

1. Ackley, D.H., Hinton, G.E., Sejnowski, T.J. (1985) A Learning Algorithm for Boltzmann Machines, *Cognitive Science*, 9: 147-169.

2. Angeline, P.J. & Pollack, J.B. (1992) The evolutionary induction of subroutines. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, 236-241. Lawrence Erlbaum
3. Bourguin, P. & Varela, F.J. (1992). Toward a Practice of Autonomous Systems. In Varela, F.J. & Bourguin, P. (Eds.) *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, (pp. xi-xvii). Cambridge, Mass.: MIT Press
4. Davies, A.P., Watson, R.A., Mills, R., Buckley, C. L., Noble, J. (2010) If you can't be with the one you love, love the one you're with: How individual habituation of agent interactions improves global utility. *ALIFE XII*, 659-666.
5. Gross, T. & Sayama, H. (2009) *Adaptive Networks. Theory, Models and Applications*. Series: Understanding Complex Systems. Springer-Verlag: Berlin.
6. Hebb, D.O. (1949) *The organization of behaviour*. New York: Wiley.
7. Hart & Mas-Colell (2000) A Simple Adaptive Procedure Leading to Correlated Equilibrium, *Econometrica*, 68, 1127-50
8. Hopfield, J.J. (1982) Neural networks and physical systems with emergent collective computational abilities, *PNAS USA*, 79 (8) 2554-2558.
9. Hopfield, J.J. Tank, D.W. (1986) Computing with neural circuits: A model. *Science* 233: 625-633
10. Hopfield, J.J., Tank, D.W. (1985) 'Neural' computation of decisions in optimization problems. *Biol Cybern* 52:141-152.
11. Kirchner, M. & Gerhart, J. (1998). Evolvability. *Proc. Natl. Acad. Sci. USA*, 95:8420-8427
12. Lewis, M. (2009) *An Investigation Into The Evolution Of Relationships Between Species In An Ecosystem*. MSc Dissertation, ECS, University of Southampton.
13. Maynard Smith, J. & Szathmari, E. (1995) *Major Transitions in Evolution*. W. H. Freeman
14. Michie, D. (1968) "Memo Functions and Machine Learning," *Nature*, No. 218, pp. 19-22.
15. Mills, R. (submitted) *Multi-scale search, modular variation, and adaptive neighbourhoods*.
16. Mills, R. (2010) *How Micro-Evolution Can Guide Macro-Evolution: Multi-Scale Search via Evolved Modular Variation*. PhD thesis, ECS, University of Southampton.
17. Parter, M., Kashtan, N., Alon, U. (2008) Facilitated Variation: How Evolution Learns from Past Environments to Generalize to New Environments. *PLoS Comput Biol* 4(11): e1000206.
18. Pelikan, M., Goldberg, DE. & Lobo, F. (1999) *A Survey of Optimization by Building Probabilistic Models*. IlliGAL-99018
19. Rosenbloom, P. S., & Newell, A. The chunking of goal hierarchies: A generalized model of practice. In *Machine Learning: An Artificial Intelligence Approach, Volume II*, R. S. Michalski, et al, Eds., Morgan Kaufmann Publishers, Inc., Los Altos, CA, 1986.
20. Santos, F.C., Pacheco, J.M. & Lenaerts, T. (2006). Cooperation prevails when individuals adjust their social ties. *PLoS Comput Biol*, 2(10), e140: 1284-1291
21. Simon, H.A. (1969) *The Sciences of the Artificial*. (MIT Press, MA).
22. Watson R.A., Palmius, N., Jackson, A., Mills, M., Powers, S., Buckley, C.L., & Penn, A.S. (in prep) *A Model of Evolved Transitions in the Evolutionary Unit*.
23. Watson, R.A., Buckley, C. L. and Mills, R. (2010) Optimisation in 'Self-modelling' Complex Adaptive Systems. *Complexity*. (in press)
24. Watson, R.A., Mills, R., Buckley, C.L., Davies, A.P. (2010) Associative memory in gene regulation networks, *ALIFE XII*, 194-202.
25. Watson, R.A., Mills, R. & Buckley, C. L. (2010) Global Adaptation in Networks of Selfish Components: Emergent Associative Memory at the System Scale. *Artificial Life* (in press).
26. Watson, R.A., Mills, R. & Buckley, C. L. (2011) Transformations in the Scale of Behaviour and the Global Optimisation of Constraints in Adaptive Networks, *Adaptive Behavior*, in press.
27. Watson, R.A., Palmius, N., Mills, R., Powers, S.T., & Penn, A.S. (2009) Can Selfish Symbioses Effect Higher-level Selection? Kampis et al (Eds.): *ECAL, Part II, LNCS 5778*, pp. 27-36, 2011.
28. Young, HP. (2001) *Individual Strategy and Social Structure: An Evolutionary Theory of Institutions*, Princeton

# Spatial structure creates community-level selection for nutrient recycling

Hywel T.P. Williams<sup>1</sup>, Richard A. Boyle<sup>1</sup> and Timothy M. Lenton<sup>1</sup>

<sup>1</sup> College of Life and Environmental Sciences, University of Exeter, Stocker Road, Exeter EX4 4QD, UK  
h.t.p.williams@exeter.ac.uk

Nutrient cycling is a ubiquitous feature of ecosystems at all scales, allowing productivity to rise beyond the limits set by external nutrient inputs. Nutrient cycling occurs as a side-effect of the metabolism of a diverse set of species that each performs a step in the recycling loop. Recycling loops can be large and involve many steps. At each step the possibility exists for ‘side-reactions’ in the form of species with metabolisms that consume an intermediate metabolite but do not create the product needed to complete the recycling loop. Also, at least some of the biochemical reactions in any closed recycling loop must be endergonic (energy-consuming) and thus recycling loops may be vulnerable to invasion or parasitism by species that consume intermediates but do not produce costly products needed to close the loop. The possibility of such destabilising side-reactions appears to conflict with the apparent stability and ubiquity of nutrient recycling in nature.

Here we propose that the ecosystem-level autocatalysis provided by nutrient recycling offers a productivity benefit that can be selected at the level of the biological community, provided that certain conditions are met: (1) the benefits of recycling must be localised so that they preferentially accrue to participants, (2) metacommunity structure must be such that multi-species communities can propagate intact. We use an idealised model of a simple microbial ecosystem (Boyle et al, submitted) to show that spatial structure can be sufficient to provide these conditions and allow community-level selection (Williams and Lenton, 2007a, 2008) to stabilise and promote nutrient recycling.

The model is an individual-based evolutionary simulation of a microbial community composed of three species which interact via their metabolic products. The community is distributed across multiple patches arranged in a ring topology to give an approximation to a 1D spatial environment. Each patch is internally well-mixed and connected to its neighbours on either side by a slow rate of diffusive mixing. Three chemical substrates are consumed/produced in the metabolism of the three microbial species. All species are assumed to be identical apart from their pattern of resource utilisation, i.e., no species has any competitive advan-

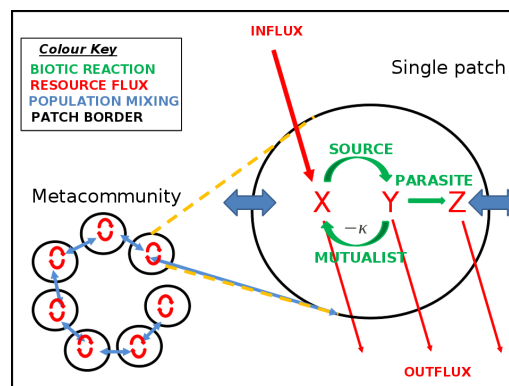


Figure 1: Patches are internally well-mixed and connected (in a ring topology) by a slow rate of diffusive mixing. Each patch is supplied with nutrient substrate X at a uniform rate. The “source” species consumes X and produces a secondary substrate Y. The “mutualist” species consumes Y and regenerates X (incurring growth rate cost  $\kappa$ ). The “parasite” species consumes Y and produces substrate Z, which is not consumed by any species.

tage other than from the relative availability of their respective metabolic substrate. The “source” species consumes substrate X and produces substrate Y as a waste product. The “mutualist” species consumes substrate Y and regenerates substrate X as a product. The “parasite” species consumes substrate X and produces substrate Z. Since we assume that the reactions  $X \rightarrow Y$  and  $Y \rightarrow Z$  are exergonic (energy-releasing), the reaction  $Y \rightarrow X$  must therefore be endergonic. Thus the mutualist species incurs an energetic cost which we implement as a growth rate penalty  $\kappa$ . The level of  $\kappa$  at which both mutualists and parasites coexist (i.e. are equal competitors) quantifies the strength of community-level selection for recycling, since coexistence implies balanced selection pressures at the individual level (for parasites) and the community level (for mutualists).

Each patch is supplied with substrate X at a steady rate, while all material substrates are removed from each patch by a slow rate of dilution. Thus in the absence of any microbial

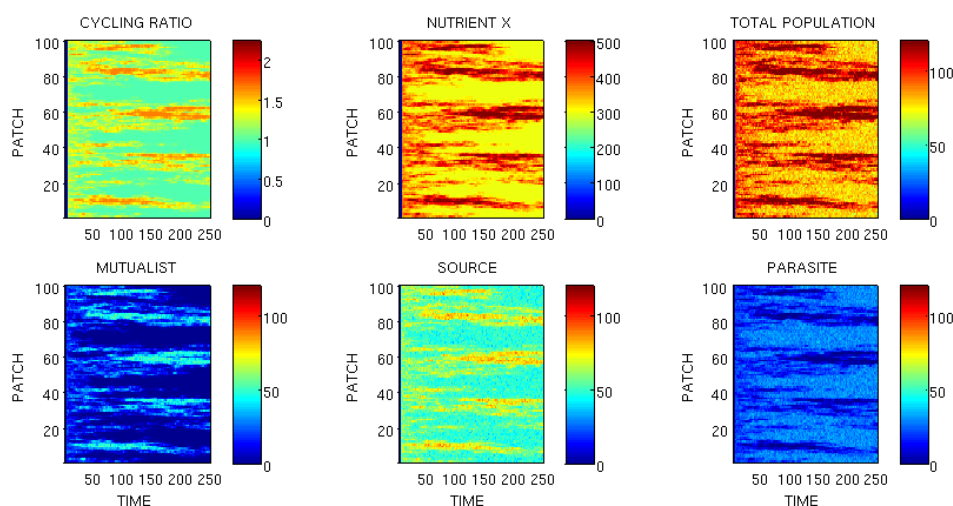


Figure 2: Example of model results showing spatial patterns in a system of 100 patches. Over time (horizontal axes) a heterogeneous distribution of species and resources over space (vertical axes) emerges from the initially homogeneous distribution. Nutrient cycling ratios are positively correlated with high density of mutualists (since this species regenerates resource X) and patch-level productivity, and negatively correlated with parasite density. Global coexistence of mutualists and parasites is stabilised by patch-level fecundity selection for recycling based on between-patch gradients in community productivity, which counteracts the within-patch advantage of the parasite.

populations the system would equilibrate with a fixed concentration of  $X$  and zero concentrations of  $Y$  and  $Z$ . The microbial community is initialised with a uniform distribution of individuals from each species. Microbes can diffuse to neighbouring patches with low probability at each timestep. There is no material mixing. Microbes grow dependent on the availability of their required substrate and reproduce by fission when their biomass reaches a fixed threshold. Microbes can die from starvation when their biomass drops below a critical threshold or stochastically with low probability (serving to represent all other causes of mortality). The system is numerically integrated using Euler's forward method.

Analytic and numerical results show that for any non-zero cost of recycling (i.e. any  $\kappa > 0$ ) parasites always exclude mutualists within a single isolated patch. Yet spatial simulations show sustained coexistence of mutualists and parasites. Mutualist frequencies in local patches are positively correlated with nutrient recycling and patch productivity. The mechanism for global coexistence of mutualists and parasites is patch-level fecundity selection; patches with higher frequencies of mutualists have higher total productivity and hence export more individuals (of all kinds) to neighbouring patches, counteracting the within-patch advantage of parasites. This is confirmed by mutualist extinction and loss of recycling when patch productivity is normalised to remove between-patch productivity gradients. Varying the spatial heterogeneity of the system by varying the between-patch mixing rate shows that recycling rates (and hence global productivity) are positively related to the 'patchiness' of the sys-

tem; low positive mixing rates that maximise spatial heterogeneity also maximise recycling. Removing spatial structure by implementing perfect between-patch mixing recovers the single-patch result of mutualist exclusion and no recycling.

The community-level selection mechanism we propose is not necessary for the formation of nutrient recycling loops in nature, which can be easily formed by aggregation of metabolically diverse species that each gain a selfish benefit from the biochemical transformations they conduct (Williams and Lenton, 2007b). However, the synergistic benefits of recycling permit community-level selection to stabilise and promote recycling, even in cases where participation incurs an individual-level cost. This finding suggests a number of testable predictions: (1) nutrient recycling should be favoured in spatially structured environments such as soils and microbial biofilms, (2) community-level productivity benefits can stabilise costly trophic mutualisms in spatially structured environments, and (3) species with complementary metabolisms should evolve traits that promote their spatial association.

## References

- Boyle, R.A., Williams, H.T.P. and Lenton, T.M. (submitted) Community-level selection of nutrient recycling in simulated microbial environment.
- Williams, H.T.P. and Lenton, T.M. (2007a) *PNAS*, 104 (21), 8918-8923.
- Williams, H.T.P. and Lenton, T.M. (2008) *PNAS*, 105 (30), 10432-10437.
- Williams, H.T.P. and Lenton, T.M. (2007b) *Oikos*, 116 (7), 1087-1105.



# Artificial Cells as Reified Quines

Lance R. Williams<sup>1</sup>

<sup>1</sup>University of New Mexico, Albuquerque, NM 87131  
williams@cs.unm.edu

## Abstract

Cellular automata were initially conceived as a formal model to study self-replicating systems. Although reproduction by biological cells is characterized by exponential population increase, no population of self-replicating machines modeled as a cellular automaton has ever exhibited such rapid growth. We believe this is due to the inability of cellular automata to model bonded complexes of *reified actors* undergoing random independent motion.

To address this limitation, we introduce a model of parallel distributed spatial computation which is highly expressive, indefinitely scalable, and asynchronous. We then use this model to define two examples of self-replicating *kinematic automata*. These machines assemble copies of themselves from components supplied by diffusion and increase in number exponentially until the supply of components is depleted. Because they are both programmable constructors and self-descriptions, we call them *reified quines*.

## Introduction

Much as Turing had done twenty years earlier when motivating his computing machine by first describing a notional human computer which computed with paper and pencil (Turing, 1936), von Neumann motivated his self-replicating machine by means of a thought experiment (Burks, 1970). von Neumann's machine assembled copies of itself from a set of components undergoing random independent motion on the surface of a lake. The components consisted of girders, hands, muscles, sensors, switches (*and*, *or* and *not* gates), and delays, together with tools for welding and cutting. von Neumann ultimately concluded that the physics of his machine was too removed from reality to be interesting, while unnecessarily complicating the study of the information processing problems inherent in self-replication. Accordingly, the bulk of his subsequent efforts were concerned with abstract machines not physical machines, and the class of abstract machine he adopted, *cellular automata*, has dominated the field for the past fifty years.

Although self-replication by biological cells is characterized by exponential population increase, no population of self-replicating machines modeled as a cellular automaton has ever displayed such rapid growth. Indeed, populations

of the most fecund (Langton, 1984) grow only as a quadratic function of time. We believe this is due to the inability of cellular automata to model bonded complexes of *reified actors* undergoing random independent motion.

Random independent motion, or *diffusion*, plays a crucial role in our work. First, as in von Neumann's kinematic model, components required for self-replication are supplied by diffusion. Second, diffusion changes the length of bonds, and vital operations must wait until bonds are of sufficient length. Third, the products of self-replication are dispersed by diffusion, which is essential for exponential population growth because it prevents overcrowding.

## Quines

Self-replicating machines can be divided into two types. The *Darwinian* type contain a self-description (genotype) and replicate by both copying it (yielding a copy of the genotype) and decoding it (yielding a copy of the phenotype). In contrast, the *Lamarckian* type replicate by copying the phenotype directly. Computer *worms* are Lamarckian, while *quines* (programs written in high-level languages which print themselves) are Darwinian. Worms don't need a self-description because of the nearly unique capacity for *reflection* possessed by machine language programs running on digital computers with von Neumann architectures. Programs and data reside in the same memory; programs *are* data. In contrast, most high-level programming languages lack the capacity for reflection. It follows that quines, like biological cells, must replicate by copying and decoding self-descriptions.

## Prior Work

The prior work with goals and approach most similar to our own is that of Hutton (2007), who has developed an artificial cell with a membrane in a 2D artificial chemistry. Hutton's cell consists of a membrane formed from a ring of 14 atoms internally bisected by a string of 5 atoms which serves as a partial genome. The membrane is permeable to unbonded atoms but impermeable to bonded atoms. The entire structure is copied atom-by-atom, through the action of 39 reaction rules which define a universal chemistry. Atoms are of

6 different types and can possess up to 62 states each. The reaction rules have a very restricted form; both left and right hand sides consist of a single pair of atoms (either bonded or unbonded), and each in a specified state.

The most impressive aspect of Hutton's work is the partial genome. This is an arbitrary string of atoms which can be used to encode any reaction rule. It is translated into a bonded pair of atoms which functions as an enzyme. Because it is contained inside a membrane impermeable to bonded atoms, it is hoarded by the cell for its exclusive use. Although enzymes can (in principle) be used to replace any of the reaction rules in the artificial chemistry (the single exception presumably being the rule governing the use of enzymes), this has only been demonstrated for a single reaction rule and Hutton (2005) states that a genome 700 atoms long (and a correspondingly larger membrane), would be needed to replace the full set.

### Actor Model

Biological cells are membranes made of lipids which contain water, enzymes, and DNA. The DNA encodes the enzymes and the enzymes (in water) form metabolic pathways which collectively: 1) copy the DNA; 2) translate the DNA into enzymes; and 3) make the membrane grow and divide. In our view, biochemistry is parallel distributed computation and enzymes are *actors*. Membranes don't just concentrate and isolate enzymes, they define private *absolute address* spaces. In effect, they permit the construction of idiosyncratic biochemistries, defined by specific sets of enzymes, the descriptions of which are encoded by the cells' own DNA.

The *actor model* is a model of parallel distributed computation (Hewitt et al., 1973). An actor is a process which possesses a unique absolute address. Using these addresses, actors send and receive messages to and from other actors. In response to receiving a message, actors can change state, create new actors, and send new messages. Significantly, and unlike cellular automata, computation in the actor model is event-driven and asynchronous.

With respect to the goal of constructing reified quines, the actor model has a number of shortcomings. First, because of its use of absolute addresses, it is not indefinitely scalable; in an actual implementation, the average time required to deliver a message increases as the number of actors increases. Second, there is no satisfactory method to generate guaranteed unique addresses in a parallel distributed manner. Third, and most significantly, the actor model is not reified—actors exist in an abstract space, not in a space which is isomorphic to physical space.

### Reified Actor Model

Although as originally conceived, actor models are not reified, it is possible to create a reified actor model or *movable feast* (Ackley and Cannon, 2011). In a movable feast, all

actors have unique positions on a 2D grid. Actors possess a finite number of states and can sense and change the positions and states of actors in their  $n \times n$  neighborhoods. Significantly, actors can create *bonds* with other actors in their  $n \times n$  neighborhoods. Bonds are *relative addresses* which are short, symmetric, and automatically updated as actors undergo random independent motion (restricted by the lengths of bonds).

The set of actors reachable through a sequence of bonds of length less than or equal to  $k$  comprise an actor's *bond graph  $k$ -neighborhood*. Actors can sense and change the positions and states of actors in their bond graph  $k$ -neighborhoods.

Like conventional actor models, computations in a movable feast are event-driven and asynchronous. Unlike conventional actor models, movable feast computations are based on the application of graph rewrite rules possessed by individual actors to the actors' bond graph  $k$ -neighborhoods. Sets of related graph rewrite rules are grouped into *behaviors*, which are indivisible and conferred as units. Actors can *possess* multiple behaviors but can *denote* at most one behavior. Significantly, an actor can confer the behavior it denotes on other actors through bonds. The distinction between possessing and denoting mirrors the phenotype-genotype distinction in biological cells and the program-data dichotomy in quines.

The update scheme in the movable feast consists of picking an actor at random, picking a behavior possessed by that actor at random, and applying the first graph rewrite rule with a pattern matching the actor's bond graph  $k$ -neighborhood.

### Kinematic Automata

The vertices of a *bond graph* are actors and the edges are bonds; both actors and bonds can be of one or more types. Because they are reified, actors have unique positions on a 2D grid. In homage to von Neumann, we define a *kinematic automaton (KA)* to be a set of reified actors possessing type specific behaviors assembled in a bond graph.

A *description* of a KA consists of a bond graph and a *behavior graph*. The behavior graph represents the relation between the set of types and the set of behaviors, *i.e.*, the *behavior relation*. Actors are finite state machines with transition functions defined by the behaviors they possess (Fig. 1). It follows that a KA is an asynchronous network of communicating finite state machines (Brand and Zafropulo, 1983); the set of behaviors possessed by its actors define a graph rewriting system (Klavins et al., 2004) which transforms the embedding and topology of the network over time.

A *programmable constructor* for a class of KA's is a KA which takes a description of a KA in the class and builds it. Example classes are *reified-strings* and *reified-sets*. A programmable constructor may (or may not) be in the class it builds. A *self-description* is a KA where the bond graph represents the behavior graph using an encoding scheme; it is

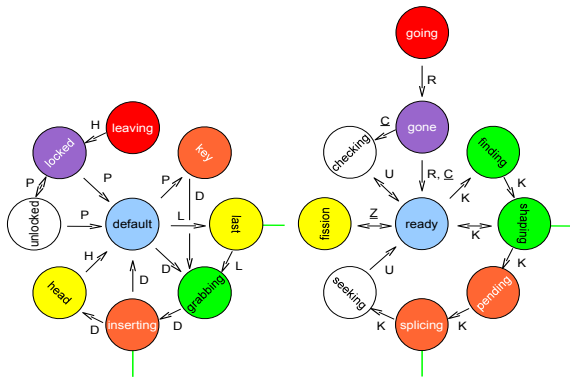


Figure 1: State transition diagrams for actors in reified-string (left) and reified-set (right) quines. Letters denote behaviors mediating state transitions. Green sticks mark states where the actor possesses a *hand* bond.

this use of dual meaning which resolves the seeming paradox of self-description—how can a thing contain a description of itself?

## Reified-String Quine

A *reified-string* is a KA consisting of a chain of reified actors linked by bonds. Apart from the head (tail) each actor in the reified-string has a unique predecessor (successor) to which it is bonded by a *prev* bond (*next* bond). A behavior graph can be represented using an adjacency list representation which in turn can be represented as a string. For example, let  $H$ ,  $D$ ,  $P$ , and  $L$  be types denoting behaviors and let  $\#$  be a punctuation type, then  $Q = \#HDP\#DDP\#PDP\##HDP\#LDPLL$  is a reified-string self-description where actors of all types possess behaviors  $D$  and  $P$  while actors of type  $\#$  also possess behavior  $H$  and actors of type  $L$  also possess behavior  $L$  (the repeated  $L$  marks the end of the string). A reified-string self-description which is also a programmable constructor for the class of reified-strings is a *reified-string quine*.

## Behaviors

To build a reified-string quine we must define a set of graph rewrite rules which when grouped into behaviors  $H$ ,  $D$ ,  $P$ , and  $L$  yield a  $Q$  which is a programmable constructor for reified-strings:

- $H$  – initiates decoding phase using *tip*
- $D$  – copies string using *grab*, *insert* and *transport*
- $P$  – confers type specific behaviors by decoding string using *key*, *lock*, *unlock* and *confer*
- $L$  – initiates copying phase, assembles daughter, and effects fission using *cleave*.

The reified-string quine copies itself in two phases. During the *copying* phase, the bond graph is copied actor-by-actor. During the *decoding* phase, the adjacency list representation

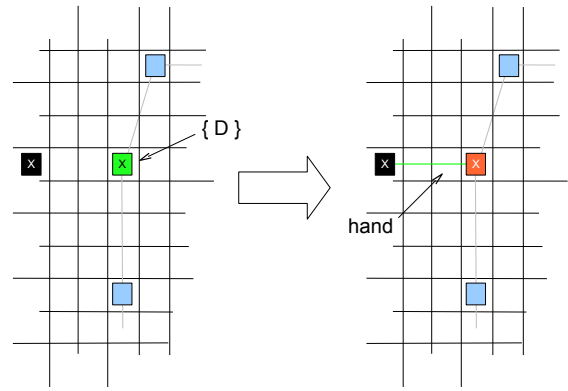


Figure 2: *Grab* graph rewrite rule. An actor in the *grabbing* state possessing behavior  $D$  and denoting behavior  $X$  forms a *hand* bond with an unbonded actor denoting the same behavior in its  $n \times n$  neighborhood. It then enters the *inserting* state.

of the behavior graph is decoded, conferring the behaviors specific to each type on the copies.

## Copying

Copying begins at the tail of the reified-string and advances towards the head. *Grab* and *insert* rewrite rules from behavior  $D$  cause each actor to

- form a *hand* bond to an unbonded actor of matching type in its  $n \times n$  neighborhood (Fig. 2)
- set that actor's state to *leaving*
- insert it into the reified-string nearer the head (Fig. 3).

In effect, the hand advances towards the head as each actor in the mother cycles through the *default*, *grabbing* and *inserting* states. Meanwhile, the *transport* graph rewrite rule from behavior  $D$  swaps actors in the *default* state with actors nearer the tail in the *leaving* state, an action which quickly moves them to the head. At the completion of the copying phase, the copied actors (which will eventually comprise the daughter) form a reversed chain in the *leaving* state attached to the mother's head.

## Decoding

The *tip* graph rewrite rule from behavior  $H$  (possessed only by actors of type  $\#$ ) recognizes when the head actor has been copied and begins the decoding phase, implemented by graph rewrite rules from behavior  $P$ . In the decoding phase, the reified-string is interpreted as an adjacency list representation of the behavior graph. This is accomplished as the copied actors traverse the mother a second time (in the reverse direction). During this traversal, each actor has its type specific behaviors conferred on it. The *key* rewrite rule causes actors denoting behaviors adjacent to actors of type  $\#$  to enter the *key* state. Actors in the *key* state *unlock* adjacent actors of matching type in the *locked* state while

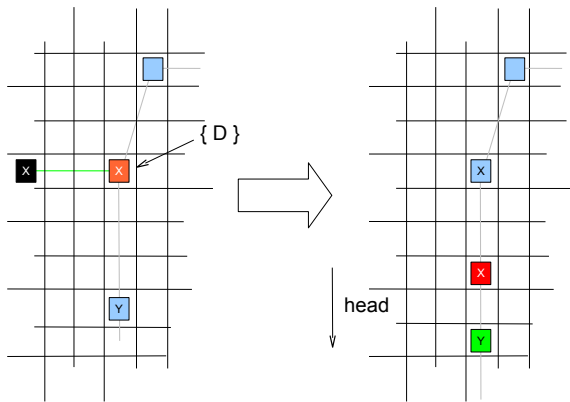


Figure 3: *Insert* graph rewrite rule. An actor in the *inserting* state possessing behavior *D* waits until its *prev* bond is of maximum length. It then inserts the actor at the end of its hand into the reified-string by bisecting the *prev* bond and enters the *default* state. The inserted actor's state is changed to *leaving* and the state of the actor previously at the end of the *prev* bond (and nearer the head) is changed to *grabbing*.

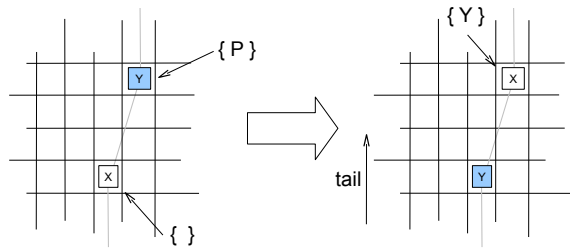


Figure 4: *Confer* graph rewrite rule. An actor in the *default* state possessing behavior *P* and denoting behavior *Y* confers behavior *Y* on the actor nearer the head when that actor is in the *unlocked* state. It then exchanges position with it, moving it towards the tail.

actors in the *default* state *confer* the behaviors they denote on adjacent *unlocked* actors (Fig. 4). Finally, actors in both *locked* and *unlocked* states are moved towards the tail.

The daughter's actors, now possessing their full complement of behaviors, are assembled into a complete reified-string at the end of a *hand* bond at the mother's tail by graph rewrite rules from the *L* behavior. When an actor in the *last* state sees two others denoting the same behavior as itself at the end of its hand, it sets both its own state and that of the nearer of the two to *grabbing* and deletes its hand (Fig. 5). This separates mother and daughter reified-strings and initiates the process of self-replication in each.

## Reified-Set Quine

In the reified-string quine, the behavior graph was encoded using an adjacency list representation, which is capable of representing arbitrary graphs. However, the reified-string quine's behavior graph was far from general—two behaviors

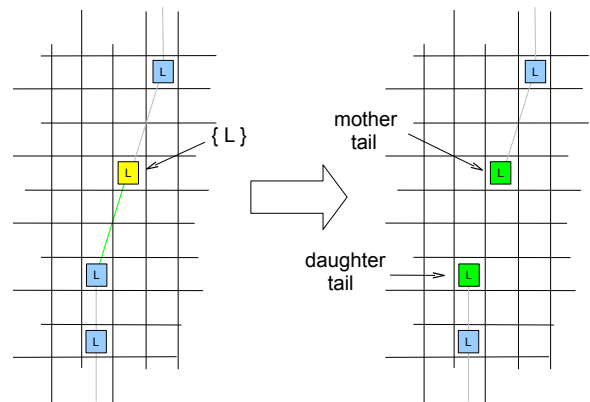


Figure 5: *Cleave* graph rewrite rule. When an actor in the *last* state sees two others denoting the same behavior as itself at the end of its hand, it sets both its own state and that of the nearer of the two to *grabbing* and deletes its hand.

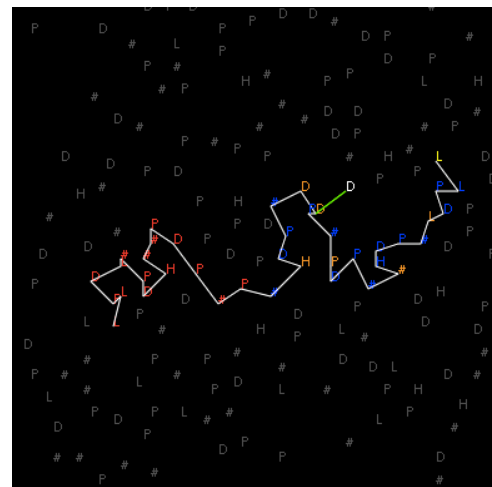


Figure 6: Reified-string quine with *hand* bond (drawn green) in the middle of the *copy* phase. Letters indicate actor type and colors indicate actor state.

(*D* and *P*) were possessed by all actors while the remaining behaviors (*H* and *L*) were possessed by only a single actor each. If we restrict ourselves to behavior relations comprised solely of *generic* behaviors and *specialized* behaviors, a more compact encoding scheme can be used.

A *reified-set* is a KA consisting of a ring of reified actors linked by *prev* and *next* bonds. A reified-set self-description which is also a programmable constructor for the class of reified-sets is a *reified-set quine*.

Reified-set quines have one great advantage when compared to reified-string quines, namely, the order of the actors in the ring is unimportant. More precisely, there is an equivalence class of bond graphs which encode a given behavior relation. Because actors can swap positions without changing the encoded behavior relation, they can possess a behavior which continually mixes their positions in the ring, ensuring that any two actors will eventually be adjacent. This

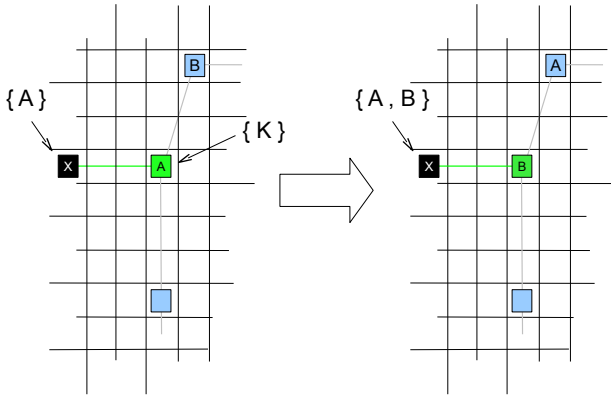


Figure 7: *Shape* graph rewrite rule. An actor in the *shaping* state possessing behavior  $K$ , when adjacent to its *continuation*, confers the behavior denoted by its continuation on the actor at the end of its hand. It then exchanges position with its continuation and leaves it in the *shaping* state.

permits a much more expressive form of parallel distributed computation than was possible with the reified-string. Indeed, if each actor in the reified-set possesses a unique *address* and a unique *continuation* then the reified-set can execute sequential programs which perform one operation for every actor. In our work, an actor's address is just the name of the behavior it denotes and its continuation is the name of another behavior. In effect, the reified-set, implemented within a *reified* actor model using *relative* addressing, can simulate a conventional *non-reified* actor model with a small *absolute* address space.

### Behaviors

Let  $X$  denote a generic behavior and  $\underline{X}$  denote a specialized behavior then  $Z = \{\underline{C}, K, U, S, N, R, M, \underline{Z}\}$  is a reified-set quine with the following behaviors:

- $\underline{C}$  – *create* daughter pinch
- $K$  – *find* matching actor, confer type specific behaviors using *shape*, then *splice* it into the reified-set
- $U$  – *seek* continuation
- $S$  – *swap* positions with adjacent actor
- $N$  – *nothing*
- $R$  – *ratchet* actors past *pinch* bonds
- $M$  – *minimize* bending energy (Williams and Shah, 1992)
- $\underline{Z}$  – *fission*.

While the reified-string quine copied itself in two consecutive phases, the reified-set quine copies itself using processes called *copy-decode*, *export*, and *verify* running concurrently in mother and daughter *subbrings*.

### Copy-decode

Copy-decode is implemented by graph rewrite rules from behaviors  $K$  and  $U$ . It is a sequential program of sixty four

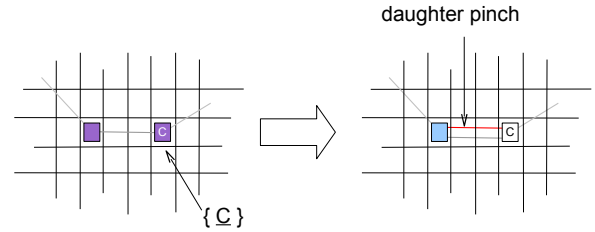


Figure 8: *Create* graph rewrite rule. An actor possessing and denoting behavior  $\underline{C}$ , in the *going* state, when adjacent to another actor in the same state, forms a *pinch* bond with the adjacent actor and enters the *checking* state (initiating the *verify* program in the daughter subring). The state of the adjacent actor is set to *ready*.

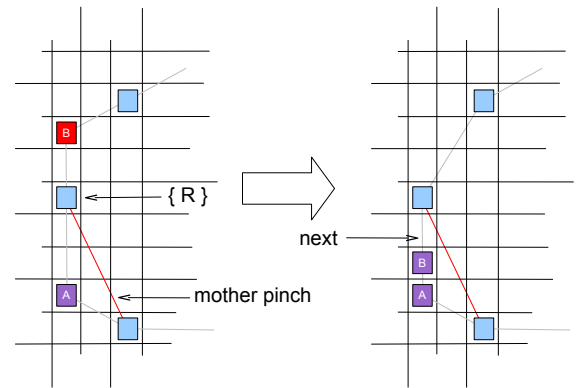


Figure 9: *Ratchet* graph rewrite rule. When at the front end of the mother's *pinch* bond, and adjacent to an actor in the *going* state, an actor with behavior  $R$  waits until its *next* bond is of maximum length. It then moves the adjacent actor past the *pinch* bond by bisecting the *next* bond, leaving the exported actor in the *gone* state.

steps which runs in the mother subring and is comprised of two nested loops—the outer loop copies the bond graph and the inner loop decodes the set representation of the behavior graph. Both loops iterate over the eight actors in the reified-set. The outer loop begins when an actor in the *finding* state:

- forms a *hand* bond to an unbonded actor of matching type in its  $n \times n$  neighborhood
- gives the daughter actor the name of its continuation (it will be the name of the daughter actor's also)
- enters the *shaping* state.

An actor in the *shaping* state waits for its continuation to be adjacent. When this happens, the actor:

- confers the behavior denoted by its continuation on the daughter actor at the end of its hand (Fig. 7)
- swaps positions with its continuation (leaving it in the *shaping* state)
- enters the *pending* state.

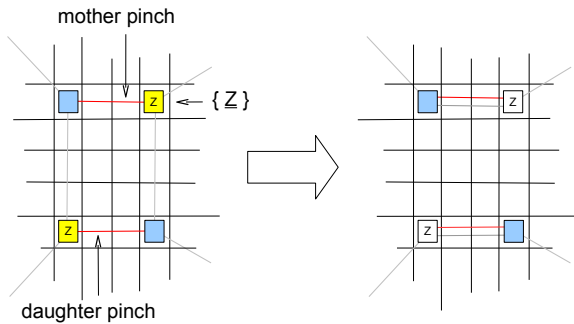


Figure 10: *Fission* graph rewrite rule. An actor possessing and denoting the behavior  $\underline{Z}$ , when in the *fission* state and located at either end of a *pinch* bond, looks for a second actor in the *fission* state in its bond graph  $k$  neighborhood in the other subring. If one exists, the *prev* and *next* bonds joining mother and daughter are rerouted so they coincide with the mother and daughter *pinch* bonds; the actors in the *fission* state enter the *seeking* state.

This begins the next iteration of the inner loop. The inner loop continues until an actor in the *shaping* state finds itself adjacent to its *pending* continuation. When this happens, the inner loop exits and the actor enters the *splicing* state. An actor in the *splicing* state waits until its *next* bond is of maximum length. When this happens, the actor:

- inserts the actor at the end of its hand into the reified-set by bisecting the *next* bond (leaving it in the *going* state)
- enters the *seeking* state.

This begins the next iteration of the outer loop. When an actor in the *seeking* state possessing and denoting behavior  $\underline{Z}$  finds itself adjacent to its *pending* continuation the copy program has finished, and the actor enters the *fission* state. It remains in the *fission* state until the *verify* process (running in the daughter subring) also completes.

## Export

Export is implemented by a set of graph rewrite rules from behaviors  $S$ ,  $R$  and  $\underline{C}$  which run concurrently with *copy-decode* and *verify* in both mother and daughter subrings. The *swap* graph rewrite rule swaps actors in the *ready* state with actors in posterior positions; a second rule portages actors around actors with *hand* bonds. These rules serve two purposes. First, they continually mix the positions of the actors in both the mother and daughter subrings, ensuring that any two actors in the same subring will eventually be adjacent. This is necessary for the *copy-decode* and *verify* programs to make progress. Second, they cause actors in the *going* state in the mother subring to move towards the gate formed by the mother and daughter *pinch* bonds—bonds created by the single rewrite rule in behavior  $\underline{C}$  (Fig. 8).

Graph rewrite rules from behavior  $R$  control a gate formed by a pair of parallel *pinch* bonds which separate the mother and daughter subrings. Another graph rewrite rule swaps

pairs of actors joined by *pinch* bonds. This routes actors in the subrings across the pinches, effectively short-circuiting the mother and daughter subrings and ensuring that the mother's and daughter's actors cannot mix. Indeed, the only actors which can get past the mother's *pinch* bond are actors in the *going* state and they can only pass in one direction. The actor at the front end of the mother's *pinch* bond, when adjacent to an actor in the *going* state, waits until its *next* bond is of maximum length. It then moves the adjacent actor past the *pinch* bond by bisecting the *next* bond, leaving the exported actor in the *gone* state (Fig 9). Another graph rewrite rule performs a similar operation at the back end of the daughter's *pinch* bond, leaving the imported actor in the *ready* state.

## Verify

*Verify* ensures that the daughter has received the full complement of actors before fission occurs. It is implemented by graph rewrite rules grouped in behaviors  $U$  and  $\underline{Z}$ . One might think that fission could occur as soon as the actor which is copied last is imported into the daughter subring. However, because of the asynchronous nature of the *export* process, there is no guarantee that the last actor copied will be the last one imported. In fact, import order inversions are common. For this reason, a simple eight step program (one for each actor in the reified-set) is run in the daughter subring to verify that the full complement has been imported.

An actor in the *checking* state in the daughter subring waits until it finds itself adjacent to its continuation. When this happens, it enters the *ready* state and sets the state of its continuation to *checking*. The one exception is the actor representing the behavior  $\underline{Z}$ —this actor is copied last and does not seek its continuation but enters the *fission* state instead.

An actor possessing the behavior  $\underline{Z}$ , when in the *fission* state and located at either end of a *pinch* bond, looks for a second actor in the *fission* state in its bond graph  $k$  neighborhood in the other subring. If one exists, the *prev* and *next* bonds joining mother and daughter are rerouted so that they overlap the *pinch* bonds; the actors in the *fission* state enter the *seeking* state, initiating the *copy-decode* program in mother and daughter, now separate (Fig. 10).

## Discussion

In the introduction, an analogy was made between enzymes and actors, and it was suggested that the primary computational function of a cell's membrane is to create an address space within which actors can send and receive messages without interference from the actors of other cells. The analogy is compelling. However, we have deliberately avoided calling the movable feast an *artificial chemistry*. One reason for not doing so is that we are trying to achieve with *dozens* of actors what is accomplished in a biological cell by *billions* of enzymes. If we are to succeed then we cannot be too literal in our imitation of the biological cell; our goal



should be to build an airplane not a bird.

## Communication

Hutton (2007) states that the primary obstacle to constructing an artificial cell with a complete set of enzymes of the sort he has described is the unwieldiness of the vastly larger genome and membrane such a cell would require. However, a more fundamental obstacle may be the difficulty of ensuring communication between enzymes and locations where reactions need to be catalyzed.

Do the enzymes of an artificial cell need to be confined within a 2D space bounded by a 1D membrane? Or can they comprise the membrane itself? Both approaches isolate a cell's enzymes from those of other cells. The second has the advantage that a simple mixing behavior guarantees communication between enzymes and locations where reactions need to be catalyzed.

## Modularity

All quines are grounded in terms defined externally in the host programming language. A programming language can have terms which are elementary and general (like Lego bricks) or complex and highly specialized (like stereo components). The terms can have uniform interfaces (like USB devices) or interfaces which limit reuse (like the pieces of a jigsaw puzzle).

The terms comprising the genome of the reified-set quine are behaviors defined outside the quine itself. A crude upper bound on the number of reified-set quine genomes would be  $2^B$  where  $B$  is the number of behaviors. Of course  $B$  can be made arbitrarily large initially, but wholly new behaviors cannot evolve; evolution is limited to discovering viable combinations of pre-existing behaviors.

Do these exist? Are there viable and interestingly different reified-set quines near  $Z$  in genome space? In partial answer to this question, we have constructed two additional examples of reified-set quines which use very different strategies to ensure that the daughter cell has received its full complement of actors. The first,  $X$ , accomplishes this by running a second instance of *copy-decode* inside the daughter subring instead of *verify*. In effect, the daughter demonstrates its viability by constructing the granddaughter. The second,  $Y$ , uses a modified *copy-decode* program which waits until it sees the most recently copied actor in the daughter subring (through the pinches) before it continues.

All three reified-set quines share behaviors  $K$ ,  $S$ ,  $R$  and  $M$  while two ( $X$  and  $Z$ ) also share  $U$ . This demonstrates that behaviors can possess a degree of modularity and potential for reuse and can be mixed and matched meaningfully. While the three reified-set quines were designed and did not evolve, the fact that they exist suggests that a future system more like Hutton (2007), with a genome containing reified descriptions of graph rewrite rules subject to mutation, would explore a genome landscape populated by viable

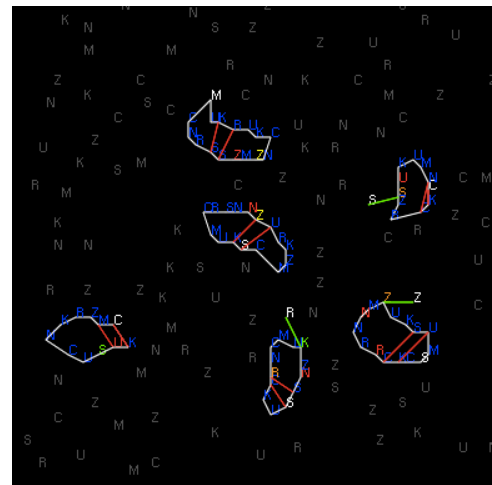


Figure 11: Six reified-set quines. Letters indicate actor type and colors indicate actor state. In the mother subring of the topmost quine, the *copy-decode* program has completed, while the *verify* program is still running in the daughter subring. *Hand* and *pinch* bonds are drawn green and red.

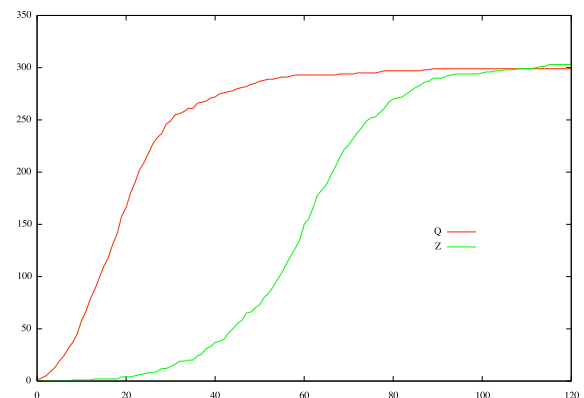


Figure 12: Exponential growth of non-competing populations of reified-string quines,  $Q$ , and reified-set quines,  $Z$ .

and interestingly different artificial cells.

## Experimental Results

In each of the three experiments, approximately 11000 unbonded actors were randomly placed on a grid of size  $512 \times 512$  to achieve a 4% area density. Except for pairs joined by *prev* or *next* bonds, actors were excluded from  $5 \times 5$  neighborhoods surrounding other actors. The maximum bond length equaled 4, the diffusion constant equaled 0.5, and search neighborhoods were of size  $11 \times 11$ .

In the first experiment, the unbonded actors were of types comprising the genomes of the  $Q$  reified-string quine and the  $Z$  reified-set quine. The proportion of each type matched that of the two genomes. A single reified-string quine and a single reified-set quine were then placed in the grid, after which, populations of both increased exponentially, in the



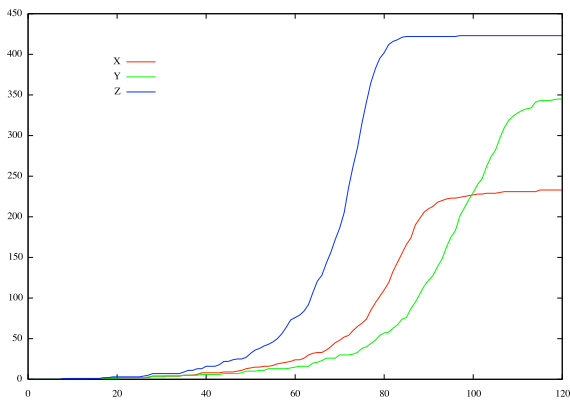


Figure 13: Exponential growth of three non-competing populations of reified-set quines. Z is the quine described at length in this paper while X and Y use alternative strategies to verify that the daughter has received its full complement of actors.

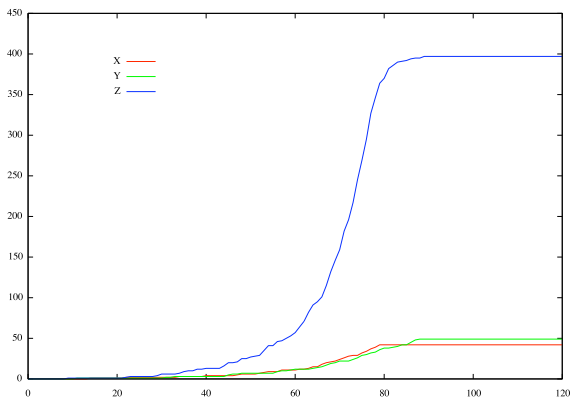


Figure 14: Three populations of reified-set quines compete for a shared resource, the *K* behavior. The Z quine outcompetes the X and Y quines.

process converting essentially all unbonded actors into approximately 300 copies of each quine (Fig. 12).

In the second experiment, the unbonded actors were of types comprising the genomes of the X, Y, and Z reified-set quines. As before, the proportions of each type matched those of the genomes; types common to all three, *e.g.*, *K*, were three times as numerous as unique types, *e.g.*, *U*. Single X, Y, Z reified-set quines were then placed in the grid. Populations of all three increased exponentially, yielding 424 copies of the Z quine, 345 copies of the Y quine, and 233 copies of the X quine (Fig. 13). The differences in these numbers can be attributed to the fact that (after all unbonded actors have been consumed) the final population consists of a mixture of individuals at various points in the self-replication process and which therefore exhibit a range of sizes. The Z quine is the most efficient at converting unbonded actors into copies of itself while the X quine is the least. This is presumably due to the fact that the X quine

requires its daughters to demonstrate their viability by constructing granddaughters. Consequently, the average size of X quine instances is significantly larger than the average size of Y or Z quine instances.

The conditions of the third experiment were nearly identical to those of the second except that the number of unbonded actors of type *K* (common to all three genomes), was reduced by a factor of three. Consequently, populations of X, Y, and Z quines were forced to compete for the under-represented shared resource. The winner of the competition was the Z quine, which succeeded in constructing nearly 400 complete individuals, while the X and Y quines succeeded in constructing less than 50 each (Fig. 14).

## Conclusion

A highly expressive, indefinitely scalable, and asynchronous model of parallel distributed spatial computation has been introduced and used to define a series of self-replicating kinematic automata. These machines assemble copies of themselves from components supplied by diffusion and increase in number exponentially until the supply of components is depleted. Because they are both programmable constructors and self-descriptions, we call them *reified quines*.

## Acknowledgements

Thanks to Dave Ackley for many helpful conversations.

## References

- Ackley, D. H. and Cannon, D. C. (2011). Pursue robust indefinite scalability. In *Proc. HotOS XIII*, Napa Valley, CA, USA.
- Brand, D. and Zafiropulo, P. (1983). On communicating finite-state machines. *J. ACM*, 30:323–342.
- Burks, A. (1970). von Neumann’s self-reproducing automata. In Burks, A., editor, *Essays on Cellular Automata*, pages 3–64. University of Illinois Press.
- Hewitt, C., Bishop, P., and Steiger, R. (1973). A universal modular actor formalism for artificial intelligence. In *IJCAI*, pages 235–245.
- Hutton, T. J. (2005). Replicators that make all their own rules. In *Proc. Workshop on Artif. Chem. and Its App., 8th European Conf. on Artif. Life*.
- Hutton, T. J. (2007). Evolvable self-reproducing cells in a two-dimensional artificial chemistry. *Artif. Life*, 13:11–30.
- Klavins, E., Ghrist, R., and Lipsky, D. (2004). Graph grammars for self assembling robotic systems. In *IEEE Conf. on Robotics and Automation*.
- Langton, C. (1984). Self-reproduction in cellular automata. *Physica D*, 10:135–144.
- Turing, A. (1936). On computable numbers, with an application to the Entscheidungs problem. *Proc. London Math. Soc.*, 2(42):230–265.
- Williams, D. and Shah, M. (1992). A fast algorithm for active contours and curvature estimation. *CVGIP: Image Understanding*, 55(1):14–26.

# Life Requires Genetic Representation and *vice versa* – Consequences for ALife

Peter R. Wills<sup>1</sup>

<sup>1</sup>Department of Physics, The University of Auckland, Private Bag 92019, Auckland 1142, New Zealand  
p.wills@auckland.ac.nz

## Abstract<sup>†</sup>

The production of autonomously functioning, integrated, complex networks of physico-chemical processes requires the creation of some mode of informational representation in molecular form (genes), not only as a matter of fact but also as the only plausible way of designing such systems to achieve control with a level of specificity typical of molecular biological processes. Likewise, only through their natural selection as parts of systems which express the information in them could DNA sequences of kilo-, mega- or giga-base length attain specific representational meanings of biological significance. Nothing worthy of the designation “Artificial Life” will exist until an information-interpreter/constructor coupling of the sort that emerged at life’s origin on our planet is recapitulated in the laboratory. Attempts to achieve such a goal require very careful scrutiny and the ethics of such endeavours should be discussed within the context of a radical critique of how human agency is constituted and how it is linked to fundamental biological processes.

## Introduction

The central thesis of Schrödinger (1944) concerning the question “What is Life?” was that the processes of biological inheritance require that information be stored in some stable, replicable, microscopic array which he chose to describe as a “quasi-periodic crystal”, using a term first coined by Delbrück (Timoféeff-Ressovsky *et al.*, 1935). Schrödinger reasoned that finely differentiated characteristics of large organisms, such as the Habsburger Lippe, that could be genetically transmitted across generations spanning centuries, must be encoded in some structural feature of the chromosomal material of an individual cell. Although his speculations concerning the atomic form of the genetic representation of heritable information were quite wide of the mark, the elucidation of the quasi-periodic linear polymeric structure of base-paired double-stranded DNA (Watson and Crick, 1953) is reasonably interpreted as a confirmation of Schrödinger’s hypothesis.

In the ensuing decades molecular biologists have uncovered in exquisite detail, and continue to do so, the ways in which the autonomous operation and maintenance of individual cells and multi-cellular organisms depends on the expression of genetic information. The paradigm of genetic expression is the biochemical control of basic metabolism that is achieved through protein synthesis. The genetic code defines a one-to-one mapping from specific base sequences of nucleic acids to corresponding amino acid sequences of proteins, which then

serve as catalysts for the multitude of elementary chemical transformations that must be effected for a cell to survive.

In spite of the success of molecular biology, very little attention has been paid to understanding, within the definitively empirical context of the discipline, some of the deeper theoretical problems presented by the idea of natural systems coming to contain an abstract self-representation in physical form. To say that the pairing of DNA sequences with the systems in which they occur is achieved through Darwinian natural selection is to beg the question “Why do some DNA-system pairings allow the generation of living organisms and others not?” Physically pairing the *E. coli* genome with a DNA-free human (stem) cell, or *vice versa*, does not produce a viable result; however, replacing the genome of *M. capricolum* with that of *M. mycoides* does (Gibson *et al.*, 2010). Thus, in spite of the many similarities in the molecular biological processes operating in different organisms, such as the near universality of the informatic rules of protein synthesis, cells cannot be construed as containing a universal constructor of the sort considered by von Neumann (1949) in his theory of self-reproducing automata. The quasi-Platonic mathematical space of genetic sequences in which all possible organisms are defined, as envisioned by neo-Darwinists (Dawkins, 1986), is an illusion. Selection of a phenotype may be a result of some arbitrary, autonomous change in the behaviour of the interpreter/constructor, rather than the result of a genetic change. A full account of biological evolution requires a description of the structural constraints that define which DNA sequences are amenable to interpretation by corresponding molecular biological systems, not just an analysis of the phylogenies of genetic sequence elements and their incidentally associated phenotypes (Wills, 2009).

This line of argument exposes the very point at which current theories of the autonomy of living systems reach their limit. The apparently general biological constructors employed by proponents of synthetic biology (Gibson *et al.*, 2010) are, in fact, virtually intact (denucleated) cells from a taxon closely related to the species of origin of the novel DNA with which the interpreter/constructor is presented. The constructor itself is a very complex system comprised of an enormous number of macromolecules, many of them specific proteins, which have only ever appeared in the cosmos as a result of their coexistence with their genetic representation. Clearly, modern biological constructors have evolved from more primitive ones. Although this process has been one of coevolution with genetic sequences, it cannot be reduced to

<sup>†</sup> The presentation of this paper at ECAL 11 in Paris is dedicated to the memory of Fernando Pereira, who was killed in Auckland on 10 July 1985 during France’s terrorist attack on our common enterprise, to which I was expert adviser, opposing the military misuse of scientific knowledge.

the evolution, through natural selection, of those sequences, as Dawkins (1986) would have it. Some means of interpreting genetic information, by way of a biological constructor, must have existed before nucleic acids of any biological value could be said to have survived as a result of natural selection.

Now that language from the theory of automata and informatics is used ubiquitously in molecular biology to describe the fundamental relationship between genetic information and the results of its expression, the chicken-egg, protein-DNA dilemma should be recast as the problem “What came first, the biological constructor or its genetic representation?” And of course the dilemma is only resolved by saying that neither precedes the other in biology, the history of which is the product of their conjoint evolution.

In this paper I will investigate the implications of this view of the origin of life as an information-constructor coupling event in relation to projects which aim to create living systems *de novo*. I conclude that our understanding of this natural coupling is so primitive that there is currently no prospect of creating true Artificial Life. I will also present a pessimistic view of the possible consequences of pursuing high-impact transformative technologies which piggy-back on the intricately elaborated intact versions of the information-constructor coupling that can be mined from extant organisms and adapted by human to the pursuit of power and ill-conceived goals.

## Biological specificity

Schrödinger (1944) enunciated the modern view of molecular biological information as a solution to the problem of explaining the stability of biological inheritance in the face of the perpetual disordering effects of microscopic thermal processes. Two decades later physico-chemical details of ribosomal protein synthesis had been elucidated and explained in terms of the existence of a genetic “code”, establishing a paradigm for the way in which genetic information is expressed in cellular systems. Although the idea of a symbolic code, a translation table between alphabets, had no precedent in the description of the physics and chemistry of natural systems, it quickly became the context of virtually all discourse about molecular biological processes. The principal theoretical expression of the new mode of description of biochemical processes was framed by two principles put forward by Crick (1958), the Sequence Hypothesis and the Central Dogma. The Sequence Hypothesis addressed a problem which was implicit in Schrödinger’s view of inheritance – the source of biological specificity, that is, what differentiates one organism from another, or one biochemical process from another, down to the level of taxon-specific molecular structures.

According to the Sequence Hypothesis (Crick, 1958) “the specificity of a piece of nucleic acid is expressed solely by the sequence of its bases and ... this sequence is a (simple) code for the amino acid sequence of a particular protein”. Then, in what we would now take as a very rough first-order approximation, the functional specificity of proteins, folded chains of amino acids, was reduced to sequence information under the assumption that “the folding is simply a function of the order of the amino acids” in the protein. Crick had obviously made simplifications that were not entirely justified

but none that caused the broad-brush picture to be abandoned as discoveries of more elaborate molecular biological processes accumulated. The Sequence Hypothesis provided the first explanation of how stably stored molecular information “got out” and had some effect in cells. Furthermore, it gave some insight into how genetic information afforded control to be maintained over internal cellular processes of metabolism and, as was discovered shortly afterwards, gene expression (Jacob and Monod, 1961). The large effects that small differences in the amino acid sequence of a protein could have on its catalytic properties clearly demonstrated the biological specificity of genetic information.

There have been many attempts, without an appeal to the existence of genetic information, to describe the appearance in the world of biochemical-like order in molecular systems. The emergent autocatalytic sets proposed by Kauffman (1986) represent perhaps the best-known systems that, in the abstract at least, meet the fundamental criterion of displaying a thermodynamically driven disorder-to-order transition in a complex network of interacting molecules. However, none of these systems has “rules” of any sort that are comparable with the quasi-cybernetic Turing-machine-like operations typical of the processes of protein synthesis and the genetic code. It is as if every new feature of these non-genetic systems emerges *de novo* from functional disorder, whereas in genetically controlled catalytic systems functional novelty can be achieved by modularizing intact subsystems whose operation is restricted to a range of variation determined by the invariant mode of their informational encoding. In fact it is difficult to envisage how the precise specificity of differentiated processes needed to define diverse individual taxa could be maintained without recourse to some kind of information whose storage system was protected from the vagaries of thermal disturbance.

The same argument can be applied to systems which store information in a combinatorial rather than a sequential fashion. As a direct consequence of their very nature, “compositional genomes” (Segré *et al.*, 2000) have very limited information storage capacity and a recent study indicates that systems employing this mode of genetic information storage do not have the capability to evolve through natural selection (Vavas *et al.*, 2010). Even if they could, their limited information storage capacity would set a low upper bound on the functional specificity that could be achieved through genetic expression. The specific nano-level control of molecular biological processes requires a very high density of information storage, such as can be achieved in the sequences of nucleic acids. That is not to say that combinatorial information is not of functional significance in biological systems, the signal transduction code described by Barbieri (2003) being a pertinent example. Rocha and Hordijk (2005) have considered these problems from a quite general perspective and concluded that any system capable of evolution requires functionally useful information to be stored in some inert form so that it can serve as a stable representation from which alternative dynamic configurations of the system can be constructed.

The argument of Schrödinger (1944) concerning the need for some system of atomic or molecular information storage is as relevant to explaining the stability of the processes that

determine and maintain biological specificity as it is to explaining the stability of biological inheritance. It seems implausible that the level of structural and functional specificity typical of molecular biological systems could be stably maintained through relationships among the available dynamic states of an autocatalytic network, dispersed as these inevitably are across spatial domains much larger than any structural features of individual molecules. The conclusion that the level of functional specificity displayed by dynamic molecular biological systems requires informational specification in a form that can be contained in a region of space even much smaller than a single cell seems inescapable. On the other hand, this does not mean that Crick (1958) gave an adequate account of the character of the processes that determine biological specificity, for he neglected altogether the thermodynamic aspect of molecular biological information processing (as opposed to information storage) that Schrödinger referred to as the “negentropy” principle and which Kauffman (1993) and others have addressed in their analyses of disorder-to-order transitions, especially in far-from-equilibrium systems (Prigogine and Nicolis, 1971).

### “What is Life?” again

Even if the Sequence Hypothesis of Crick (1958) is taken to be a heuristic device rather than being empirically testable, its simplicity and elegance obscure a deeper flaw in the picture of how cells are able to maintain themselves and reproduce. That flaw has not been corrected during the decades in which it has been discovered that the biochemical control of intracellular processes is much more complicated and elaborate than it first appeared to be. No matter how refined a description of a cell’s molecular biology may be, if it implicitly assumes that the specificity of molecular biological processes originates solely in genetic sequence information then it fails as a scientific explanation because it gives no account of the origin of the means of interpretation of the information. Following Crick, one is forced to assume that the ribosomal machinery and all of the other components of the protein synthetic apparatus, or some earlier, simpler version of it, were provided by evolution as a molecular biological “free lunch”.

The genetic meaning of any nucleic acid sequence cannot be determined except within the context of a physico-chemical system that acts as an interpreter or constructor of some sort. And the hallmark of the molecular components of cellular interpreter/constructors is their extremely refined specificity of action. Furthermore, the integrated action of a large number of components with very specific structures and interactions is needed to maintain the specificity of any one of them. This could not be achieved in the absence of genetic information, as we have just observed, but it is equally true that nucleic acid sequences would be devoid of biological meaning in the absence of integrated, functional specificity. Thus, it could be said that biological specificity originates as much in itself as in the genetic information it uses to maintain itself.

On this basis a limited definition of elementary life may be given down the following lines: *a complex, microscopic, dynamic, physico-chemical system may be said to be living if recurrent synthesis of its structurally specific molecular*

*components occurs as a result of their mutual co-existence with a store of molecular information whose interpretation is defined by the processes occurring in the system.* This suggested definition expresses the maxim given as the title to this paper, that “life requires genetic representation and *vice versa*”, but compared to the enduring view of Crick (1958) it emphasizes a quite different aspect of genetic information in biological systems.

Crick (1958; 1970) described the role of genetic information in molecular biological systems in his Central Dogma, which is most easily stated in the form “once information has got into protein it can’t get out again”. Although application of the Central Dogma was limited to the determination of the polymeric sequences of nucleic acids and proteins, its combined effect with the Sequence Hypothesis, identifying specificity with sequence information, was to create a view, still widely held among molecular biologists, of genes as the ultimate determinants of all biological specificity. And this view has been elaborated even more widely in the neo-Darwinian interpretation of evolution, according to which genetic mutation is the ultimate source of all biological novelty (Dawkins, 1989).

Contrary to this picture of living systems portrayed by the Central Dogma and neo-Darwinism, the tentative definition of elementary life provided above gives prime place to the maintenance of self-representation in genetic information as the cardinal feature of living systems, not the determinative existence of genetic information *per se*. The Central Dogma is often stated as the epithet “DNA makes RNA makes protein” under the implicit assumption that the means of information transfer are a given. Biological information transfer is taken unproblematically to have arisen as a result of molecular selection. It is conceded that the elementary molecular biological interpreter (the apparatus of protein synthesis and the code) was somehow bootstrapped into existence through a series of molecular events which remain a fascinating physico-chemical puzzle, but the possibility that information theoretic aspects of the origin of coding were the dominant constraining features of the process is seldom contemplated. However, the current enquiry leads us to redirect attention into the origin of life to focus on the emergence of an interpreter of genetic information (Wills, 2009), not its accumulation through natural selection (Eigen, 1971).

### Origin of life

According to the definition espoused in this paper, the most important feature of life’s origin is the emergence of an autocatalytic system of molecular components whose synthesis has an obligatory dependence on extant information stored in some molecular/atomic form. At first the catalytic specificity represented in such a system is likely to have been very restricted and the amount of information stored very small. One could envisage a system in which the autocatalytic set comprised no more than early representatives of the Class I and II amino-acyl tRNA synthetase (AARS) proteins whose polymeric sequences were differentiated by the specific placement of amino acids from two distinguishable classes, perhaps {glycine, alanine} and {valine, aspartic acid}. Through their combined operation, these proteins would

mutually produce themselves as a result of their individual capabilities of roughly differentiating between two classes of primitive codons, perhaps {GGC, GCC} and {GUC, GAC} (Eigen and Winkler-Oswatich, 1981), in two special genetic sequences, which may even have been complementary nucleic acid strands (Rodin and Rodin, 2006). A precursor system may have employed a one-letter code for these four amino acids (Francis, 2011). Although this is all speculation it serves to illustrate in elementary form what the proposed definition of life takes to be the prime feature of molecular biological systems – the necessity of information-processing dependent constructive autocatalysis.

It has been demonstrated that some physico-chemical systems afford, even from initial conditions comprising completely random synthetic events, the stepwise autocatalytic emergence of increasingly specific coded information processing (Wills, 2009; Füchslin and McCaskill, 2001; Markowitz *et al.*, 2006). And what is most interesting is that the obligatory facilitating feature of the emergence of coding in such systems is the satisfaction of what may be described as informatic boundary conditions (Wills, 1993; Neiselt-Struwe and Wills, 1997). These conditions amount to constraints on the complex relationship between the distribution of catalytic activities among molecular structures and the specific genetic sequences needed by an autocatalytic set of information-dependent synthetases. When the appropriate informatic boundary conditions are satisfied coding can be sustained in the presence of a genetic sequence that serves as a self-representation of a particular autocatalytic set of information-dependent synthetases. Generalizing this feature of molecular biological information-processing leads to the conclusion that any definition of life must include some description of purely formal features of correspondences between symbolic sequences, of which particular polymers and their physico-chemical properties are no more than particular instantiations.

As a simple calculation demonstrates (Wills, 1993), a genetic sequence potentially interpretable as a source of information for coding autocatalysis has virtually zero probability of coming into existence as a result of undirected competition between replicating polymers. This leaves open only one plausible path to life: an autocatalytic system directing the selection of nucleic acids whose sequences are “reflexive” (Wills, 2001) *vis-à-vis* their translation into functional form, “interpretation as self-representation”. The coupling between autocatalytic processes and the replication of information polymers necessary to effect the directed selection of meaningful genes cannot occur in a homogeneous system (Wills, 1994; Füchslin and McCaskill, 2001) and therefore some sort of spatial localization of associated molecular processes is entailed in the very notion of emergent information-processing at the origin of life. Autonomous control of such localization is germane to the definition of life given by Gánti (2003) as well as the idea of a cellular autopoietic network (Maturana and Valera, 1980).

On these grounds it seems highly implausible that nucleic acids could attain any representational meaning of biological significance except through their natural selection as components of systems, which express the information in them. If life is defined, as proposed, in terms of the information-interpreter/constructor coupling observed in

extant molecular biological systems, then this amounts to saying that genetic representation is impossible except in spatially localized living systems. Our understanding of the emergence of an information-interpreter/constructor coupling at the origin of life is still primitive, the question having been addressed only in studies by Füchslin and McCaskill (2001) and Markowitz *et al.* (2006). At least the need for a non-equilibrium phase-transition in the dynamics of systems that synthesize polymers randomly has been established; as has the way in which the complexity of the alphabet for genetic representation can increase in a stepwise manner (Wills, 2009) leading to a rapid expansion in not only the amount of information that can be stored but also the specificity of function of individual molecular components that can be maintained in such systems.

Beyond the genetic code that determines the specificity of ribosomal protein synthesis there are many other modular processes in biological systems that are amenable to the direct transfer of symbolic information. Barbieri (2003) associates the emergence of higher level codes with major transitions in the trajectory of biological evolution.

## Synthetic biology

It is widely accepted that the life of every organism, however life is to be defined, is derived from the life of its parent(s) such that all terrestrial life can be traced back to a single origin some three to four billion years ago. If we assume that organisms always contain genetic information and we allow an abbreviated definition of the life of an organism as *recurrent synthesis of its structurally specific molecular components* then we see that there is inter-generational continuity in the specificity of the controlled, microscopic, irreversible processes occurring in cells as well as inter-generational continuity (with variation) in the genetic complement of cells. Life has continued as an unbroken chain since its origin because cells acquire their complex dynamic state, as well as their genes, through the processes of biological inheritance.

According to our proposed definition of life we can take genes to be sequences of symbols rather than the physical entities, the nucleic acids, in which they are instantiated. This is not in any way at odds with the manner in which molecular biologists have thought about and manipulated genes ever since the language of a genetic “code” was first developed. In fact the process of genetic engineering consists increasingly of procedures of calculation using symbolic sequences. The process of instantiating designed genes as DNA molecules and inserting them into cells is just the very last step in the typical production of a modified organism. Physical causation is of little consequence in the whole process.

In describing their latest enterprise with *Mycoplasma*, a team at the J. Craig Venter Institute claims to have created a new taxon, also referred to as a “synthetic cell”, by starting from digitized genome sequence information (Gibson *et al.*, 2010). In making the claim that the cell is synthetic these scientists are suggesting that mental processes have been in some way causative in bringing the cell into existence. The alternative is to accept a restriction to explanations in terms of physical causation, on which basis there is no distinction between what is natural and what is artificial or synthetic.

Adopting that point of view we would say that the modified organism came into existence as a result of the firing of particular neurons in particular human brains and particular changes in particular electronic circuits. However, the same result would have been arrived at through a quasi-infinite set of similar world-lines with nothing particular in common except their completely arbitrary specification as involving the same symbolic representations of genetic sequences. So, let us first accept the team's claim to some kind of causative agency. What is the extent of their agency in the cell's coming into existence?

The team started with an intact cell of *M. capricolum*, removed its DNA and replaced it with synthetic DNA, whose sequence had been copied from *M. mycoides* and then slightly altered. If, unbeknown to them, there had been a single error in the sequence they synthesized, corresponding to a fatal mutation, then they would have had no grounds to claim the creation of a synthetic cell. That being the case, it seems illegitimate to ascribe agency to symbolic information-processing associated with matter configured in human neurological form and no agency whatsoever to matter configured in the form of *M. capricolum*. This problem is not resolved by the statement of Gibson *et al.* (2010) that “the DNA software builds its own hardware”. In fact, ascribing constructive agency to an entity comprised of symbols (software) violates the scientifically conventional continuity of physical causation. By any normal delineation between physical and symbolic entities one would have to say that it is the hardware which builds itself by reading the information in the genetic software made available to it. A cell envisaged as computer hardware that can remain operational and transform itself to new specifications when the program it is executing to maintain itself is suddenly swapped for a different one seems more worthy of the descriptions “creative” and “innovative” than members of *H. sapiens*, conceived as assemblages of molecules which effect minor changes in DNA sequences *in vitro*. However, the idea espoused in the definition of life proposed in this paper is that agency in living systems arises from neither software nor hardware but from the coupling between them that corresponds to symbolic self-representation. [It is quite usual for biologists to ascribe some sort of active agency to natural selection, offering explanations such as “selection made a change to the system that improved function” (Johnson and Lam, 2010).]

Scientific discourse is ill-equipped to start defining the nature and extent of agency entailed in the autonomous operation of living systems, because there is no agreed formal description agency that can serve as a basis for either theoretical analysis or empirical enquiry. But then, without admitting that ethics are essentially about the status and rights that are appropriate to diverse agents, scientists have little hesitation in taking their own assessments of the significance of what they have done as a context for framing discussion of ethical aspects and implications of their field of research. As Gibson *et al.* (2010) state “We have been driving the ethical discussion concerning synthetic life from the earliest stages of this work”. It would be foolish to denigrate such efforts or the deliberations behind them. However, it is difficult to see how such a discussion could acquire any worthwhile depth in the absence of a penetrating critical analysis of global institutional structures that give inordinate weight to scientific perspectives

in which agency is treated as if its existence were purely metaphorical, except when associated with humans. More concerning this shortly.

## ALife and Living Technology

In terms of the definition of living systems proposed in this paper, creating an artificial form of life will entail the construction of a never-before-seen coupling between a self-maintaining, complex, physico-chemical system and its self-representation in a store of molecular information. It would be difficult to convince this author that a system whose primary mode of information transfer resembled ribosomal protein synthesis in any significant detail could qualify as being truly artificial. This assessment is made on the grounds that the self-representational information-interpreter coupling found in terrestrial biological systems constitutes a “design”. [It is noteworthy that use of the term “design” is not being restricted to symbolic representations associated with brain states of members of *H. sapiens* and their artefacts; or those belonging to other supposed intelligences, whether they be material, purely mental, aetherial, or spiritual, however such categories might be conceived of.] The design for specifying the construction of a supposedly artificial system employing nucleic acid to protein information transfer and mimicking details of ribosomal protein synthesis, in essence the life of the constructed system, could reasonably be called “(a) property” that had been appropriated, almost entirely, from an extant living system. A cell created from homogeneous preparations of individual components to operate as an encapsulated nucleic acid-protein-ribosome system would indeed qualify as an example of the “synthetic cell” that Gibson *et al.* (2010) actually have failed to achieve, but it would not be Artificial Life because its design would have been copied from the version of life found naturally occurring on this planet.

At this point it seems relevant to ask what might motivate construction of a form of Artificial Life that truly met the criteria that have now been outlined. It is nearly two centuries since Shelley (1818) identified the motivation to cobble together organisms from dead parts as a quest to take in hand the intrinsic power of what is conceived to be the principle of life; in mythological terms, the fire of the gods stolen from Zeus by Prometheus and given to mortals. For Bedau *et al.* (2010) the harnessing of such power is implicit in the creation of technology that incorporates the most basic features of living systems. Although they do not focus exclusively on Artificial Life, they deem technology to be living “if it is powerful and useful precisely because it has the core properties of living systems, including such properties as the ability to maintain and repair itself, to autonomously act in its own interests, to reproduce, and to evolve adaptively on its own” and predict that during our lifetimes we will see “technology that is robust, autonomous, self-repairing, self-reproducing, evolving, adapting, and learning — a powerful combination of life's core properties that no current technology yet embodies” with the final assessment that “this transition will be a truly singular event in human history”.

Although they acknowledge potential dangers, Bedau *et al.* (2010) see ripe opportunities for living technology in medicine, environmental sustainability, energy cycles,

advanced materials, individually adapted manufacture, self-organising software, *etc.* They also consider it possible to initiate an evaluation of living technology without either discussing the nature of life, except in terms of a list of core properties, or tackling the problem of “design” – the relationship between physical reality and its representation on which the natural/artificial divide is founded. These authors then find it unsatisfactory that most people think about *Frankenstein* (Shelley, 1818) or *Prey* (Crichton, 2002) when they hear about protocells – rather minimal artificial life forms far simpler than the most elementary modern bacteria – without even a hint that what they are proposing might fail to respect a value that is intrinsic even to the most primitive living systems. Humanity’s construction of ethics is only now beginning to adjust to the idea that our behaviour may be an affront to norms that precede our evolutionary arrival in the cosmos. Could it be that the historical transition envisaged by proponents of Artificial Life and Synthetic Biology will play out as the encounter of *H. sapiens* with some aspect of reality of which, so far, we have only the faintest inkling, the “life” we cannot yet adequately define, but which is as fixed and immovable in its reactive behaviour as the physical aspect of reality, something we will discover to our own self-determined peril? That is what *Frankenstein* (Shelley, 1818) is about primarily, not the shocking monster.

Both Gibson *et al.* (2010) and Bedau *et al.* (2010) make it clear that the prospect of creating Artificial Life raises new questions of ethics and they appeal to well-accepted values like human health, environmental sustainability and human rights as the proper context for in-advance ethical evaluation of the emerging technology. Elsewhere, another group (Bedau *et al.*, 2009) has proposed ethical guidelines for enterprises concerned with artificial cells; and the ramifications of current activity in Synthetic Biology has been subjected to quite detailed analysis (Rabinow and Bennett, 2009), albeit from a perspective deeply imbued with the values of postmodernity (Forman, 2010). Laudable though these efforts are, none involves consideration of the possibility that the socially constructed motivation for pursuing Artificial Life, that is, the appropriation by *H. sapiens* of nature’s inherent capacity for self-construction through symbolic representation, may be misdirected in the sense that it will ultimately prove to be a mode of self-destruction rather than self-construction. How are we to judge?

The remainder of this paper can be taken as an illustrative approach to this problem, an attempt to start down a pathway that may assist to conceive of and realise a different representation of humanity’s future.

## Global, historical implications

Scientists are generally unlikely to warm to much of his philosophy, let alone his politics, but Martin Heidegger has to be credited with having set in motion many of the last century’s most profound considerations relevant to the relationship between physical reality, its representations and its utility. The early Heidegger (1939) was convinced by Aristotle’s portrayal of the real world whose processes are open to observation (*physis*; nature) as more than a succession of material states. He found the role of *techné* (technique, know-how) as a cause of change in the world to be crucial for

a proper understanding of human reality. In his discussion of Aristotle’s truism “a human being is generated from a human being, but not a bedstead from a bedstead” (since Antiphon had observed that, at most, a tree would grow from a planted wooden bedstead), Heidegger (1939) explains that there has been a historical misunderstanding of the role of *techné* in the generation of things that grow, as opposed to artefacts. Elsewhere he describes *physis* as “the realm of things that emerge and linger on” (Heidegger, 1959) conceiving of nature’s essence in terms more biological than simply physical or mechanical. According to Heidegger, our misconstruing of nature as a self-making artefact provides the ground for our mastering nature through technology and making it subject to our own narrow purposes. The later Heidegger (1977) is more concerned with the historical consequences of our technological mastery of nature and he characterizes technological society’s consciousness of the real world as *Ge-stell* (“En-framing”), a conception that leads us to treat existence, our own even, as *Bestand* (“standing reserve”), at hand, ready for use.

Whether or not one is sympathetic to Heidegger at all, he is the major figure in a philosophical tradition that can hardly be ignored by scientists finally seeking to exercise, in advance, some responsibility for actions of theirs that may have momentous historical consequences. One of the things we learn from that tradition is to question the structure of consciousness and its dependence on the vagaries and arbitrariness of internal constructs, especially as these influence and constrain our conception of nature; and that means bringing to bear rigorously, in a self-reflective fashion, considerations and critical analyses from all disciplines, especially those that challenge the complacency of the modern scientific perspective and the culture of power in which it is embedded. In respect of assessing the value of Artificial Life, we have to ask, outside of the comfort of the cultural context of our own experience, what each of the things that self-evidently has value, like health, environmental sustainability, and human rights, actually is; from what more general point of view might these things have value (and therefore justify the pursuit of ALife); whether they have anything like a “natural” connection to terrestrial life; and therefore whether these values are related to ALife in a way that may not become obvious in the process of creating the technology. The ethical issues raised by Artificial Life cannot be framed without deconstructing some of the most basic tenets of international law, global business practice and even scientific experimentation, namely, human ownership of and control over the functional processes and genetic identity of biological systems. While Rabinow and Bennett (2009) conclude their considerations of the ethical ramifications of Synthetic Biology by alluding to some of these problems, their descriptions of the relevant research activities accept proponents’ ideas of technological progress and its legitimacy rather unquestioningly.

Normative values like wealth, innovation, growth, health and security, derived from spheres of predominantly economic, medical and military activity in what is known to itself as “the developed world”, provide a poor basis for determining humanity’s relationship with complex biological systems. It is clear that science is now losing much of its previously proud independence from such norms. As Forman



(2010) shows, the moment we attempt to investigate how modern global society places science and technology in relation to its normative values, we see that science has been downgraded and technology upgraded in cultural rank. This process has occurred during the last three decades or so, the exact period during which the basic properties and functions of living systems have started to become objects of human exploitation. Artificial Life and Synthetic Biology are both positioned as primarily technological rather than scientific enterprises, with strong links to the privileged economic base of global human power (Rabinow and Bennett, 2009). The effect of the recent elevation of technology above science can be seen in the research community itself. The methodical, disinterested scientist has been displaced by the single-minded entrepreneur, who resourcefully pursues self-interest with clever disregard for apparently irrelevant aspects of prevalent codes of practice. Attention or lip-service to ethics is integrated into a system of shallow legitimization of whatever is deemed desirable and economically achievable by those who seek, or have become accustomed to, the power that shifts into the sphere of the new technology.

Rather than accepting this state of affairs and assuming that Darwinian forces operating in various institutional, social, economic and legal systems will finally determine what is of value, we are in a position to use the forces of reason and conscience to evaluate and choose what contribution we make to science and technology. There are some broad lessons to be learned by taking history as our guide, lessons that have to do largely with the state of ignorance, rather than knowledge, that obtains in any particular epoch. Humanity's current global crises are aggregations of the effects of many local actions conceived and conducted largely in the absence of any perception of their possible broader consequences. However, even scientifically informed and motivated judgments can result directly in consequences that turn out, with hindsight, to be undesirable because the science of any epoch, our own included, is limited to over-arching assumptions that cannot be guaranteed to do justice to all of reality. For example, the early Darwinian naturalist Walter Buller believed that the replacement of endemic avian species in New Zealand by superior exotic types was a foregone conclusion. He shot the already rare *huia*, now extinct, so that there would be forever preserved, in far off imperial museums, good specimens of this species, unique on account of its sexually differentiate mandibles (Buller, 1873).

In what potential ways could current conceptions of Artificial Life be proved inadequate in respect of actions, performed now, being deemed, on the basis of subsequent experience, later to have been based on profound ignorance?

### Assessing future prospects

If, as has been proposed in this paper, the cardinal feature of living systems is the self-representing ("reflexive"), information-interpreter/constructor coupling which emerged on the planet in its most primitive manifestation with the origin of the genetic code some three to four billion years ago; and if, associated with major evolutionary transitions, the establishment of further couplings of that character enabled more complex versions of biological autonomy based on the exchange of symbolically encoded information between

differentiated subsystems (Barbieri, 2003); and if the general possibility of such couplings is a purely formal feature of the cosmos, not necessarily related in any specific way directly to any detailed feature of the particular physical universe that we happen to inhabit; then, because biology lacks concepts adequate for the task of answering the question "What is life?", we are indeed profoundly ignorant of the consequences that may follow from our appropriation of the fundamental processes of biological causation in pursuit of short-term institutional or societal goals.

If there is anything that should bear the name "bioethics" then it is not to be found in the endless committee considerations of the immediate consequences of experimentation in genetic manipulation, cloning or reproductive biology, but rather in a critical enquiry into the intrinsic value which various forms of life bear relative to one another simply on account of their biology. The recent appearance of "sustainability" on the global political agenda is one of the few causes for optimism, for it demonstrates a nascent recognition of not only past human failures but also the need, for survival, to give effect to the sense that the high intrinsic value that we assign to ourselves, primarily on account of our consciousness, has an intimate connection with the much more prosaic "health of the biosphere". Seen within that context the biotechnological commodification of the terrestrial version of life might indeed be interpreted as the arrogation of value which has a natural location outside of human control.

Therefore, it would seem unwise to pursue the creation of Artificial Life simply because it is technically possible to do so and because it holds the promise of further power for *H. sapiens*. There is much evidence that humanity is ill-equipped to handle the complex distribution of power over nature and global society that technologies have already conferred, often resulting in irreversible losses due to uncontrollable processes, all uncompensated by any increase in value elsewhere. The fundamental modes of biological causation are far more obscure and convoluted than those of mechanical, thermal or nuclear technology. Seizing the power offered by "living technology" and using it to further current human interests, while failing to recognize the intrinsic value of the systems being tinkered with, is likely to result in a recapitulation of errors made in the deployment of other transforming technologies. It is singularly inappropriate that scientists whose consciousness is embedded in a privileged culture which already wields global power should create and propagate, according to their perceptions of what is of value to humanity and nature, a new mode of controlling the most fundamental processes of nature – those that make life itself possible. The scientific community owes it to the rest of global society to engage in a broad-ranging discussion concerning the disposition of the power that the new technologies of Artificial Life and Synthetic Biology make available, instead of attempting to reassure a skeptical public that their unfounded fears are guaranteed to evaporate into thin air as a result of appropriate education framed in terms of current scientific concepts. This author is one scientist who does not hold to the majority opinion that Artificial Life is a value-free scientific enterprise and wishes to endorse his work as follows.

It is the author's wish that no agency should ever derive military or purely financial benefit from the publication of this paper. Authors who cite this work in support of their own are requested similarly to qualify the availability of their results.

## Acknowledgements

The author is grateful to Luis Rocha and John McCaskill for their thoughtful comments and suggestions concerning an earlier version of this manuscript.

## References

- Barbieri, M. (2003). *The Organic Codes*. Cambridge University Press.
- Bedau, M. A., McCaskill, J. S., Packard, N. H. and Rasmussen, S. (2010). Living Technology: Exploiting Life's Principles in Technology. *Artificial Life* 16:89–97.
- Bedau, M. A., Parke, E. C., Tangen, U., & Hantsche-Tangen, B. (2009). Social and ethical checkpoints for bottom-up synthetic biology, or protocells. *Syst. Synth. Biol.* 3:65 – 75.
- Buller, W. L. (1873). *A history of the birds of New Zealand*. J. Van Voorst, London.
- Crichton, M. (2002). *Prey*, HarperCollins, New York.
- Crick, F. H. C. (1958). On protein synthesis. *Symp. Soc. exp. Biol.* 12:138–163.
- Crick, F. H. C. (1970). Central Dogma of Molecular Biology. *Nature* 227:561–563.
- Dawkins, R. (1986). *The Blind Watchmaker*. Longman, Harlow
- Eigen, M. (1971). Self-organization of Matter and the Evolution of Biological Macromolecules. *Naturwiss.* 58:465–523.
- Eigen, M. and Winkler-Oswatitsch, R., (1981). *Naturwiss.* 68:282–292.
- Forman, P. (2010). (Re)cognizing Postmodernity: Helps for Historians – of Science Especially. *Ber. Wissenschaftsgesch.* 33:157–175
- Francis, B. R. (2011). An alternative to the RNA world hypothesis. *Trends in Evolutionary Biology* 3:2–12.
- Füchslin, R. M. and McCaskill, J. S. (2001). Evolutionary self-organization of cell-free genetic coding. *Proc. Natl. Acad. Sci. USA* 98:9185–9190.
- Gánti, T. (2003). *The Principles of Life*. Oxford University Press.
- Gibson D. G., Glass, J. I., Lartigue, C. et al. (2010). Creation of a Bacterial Cell Controlled by a Chemically Synthesized Genome. *Science* 329:52–56.
- Heidegger, M. (1939). On the Essence and concept of  $\Phi\upsilon\varsigma$  in Aristotle's Physics B, I. In McNeill, W. (1998), editor, *Pathmarks*. Cambridge University Press.
- Heidegger, M. (1959). *An Introduction to Metaphysics*. Yale University Press.
- Heidegger, M. (1977). *The Question Concerning Technology and other essays*. Harper Colophon Books, New York, NY.
- Jacob, F. and Monod, J. (1961) Genetic regulatory mechanisms in the synthesis of proteins. *J. Mol. Biol.* 3:318–356.
- Johnson, B. R. and Lam, S. K. (2010). Self-Organization, Natural Selection, and Evolution: Cellular Hardware and Genetic Software. *BioScience* 60:879–885.
- Kauffman, S. A. (1986). Autocatalytic sets of proteins. *J. Theor. Biol.* 119:1–24.
- Kauffman, S. A. (1993). *The Origins of Order*. Oxford University Press.
- Nieselt-Struwe, K. and Wills, P. R. (1997). The emergence of genetic coding in physical systems. *J. Theor. Biol.* 187:1–14.
- Prigogine, I. and Nicolis, G. (1971). Biological order, structure and instabilities. *Quart. Rev. Biophys.* 4:107–148.
- Rabinow, P. and Bennett, G. (2009). Synthetic biology: ethical ramifications 2009. *Syst. Synth. Biol.* 3:99–108.
- Rocha, L. M. and Hordijk, W. (2005). Material Representations: From the Genetic Code to the Evolution of Cellular Automata. *Artificial Life* 11:189–214.
- Rodin, S. N. and Rodin, A. S. (2006). Partitioning of aminoacyl-tRNA synthetases in two classes could have been encoded in a strand-symmetric RNA world. *DNA Cell. Biol.* 25:617–626.
- Segré, D., Ben-Eli, D. and Lancet, D. (2000). Compositional genomes: Prebiotic information transfer in mutually catalytic noncovalent assemblies. *Proc. Natl. Acad. Sci. USA* 97:4112–4117.
- Schrödinger, E. (1944). *What is life?* Cambridge University Press.
- Shelley, M. W. (1818). *Frankenstein, or the Modern Prometheus*. University of Chicago Press, Chicago, IL. 1974.
- Timoféeff-Ressovsky, N. W., Zimmer, K. G. and Delbrück, M. (1935). Über die Natur der Genmutation und der Genstruktur. *Nachr. Ges. Wiss. Göttingen, Fachgruppe VI N. F.* 1:189–245.
- Vavas, V., Szathmáry, E. and Santos, M. (2010). Lack of evolvability in self-sustaining autocatalytic networks constraints metabolism-first scenarios for the origin of life. *Proc. Natl. Acad. Sci. USA* 107:1470–1475
- Von Neumann, J. (1949). In: Burks A. W. (1966), editor. *Theory of Self-Reproducing Automata*, pages 29–87. University of Illinois Press, Urbana, 1966.
- Watson, J. D. and Crick F. H. C. (1953). Genetical implications of the structure of deoxyribonucleic acid. *Nature* 171:964–967.
- Wills, P. R. (1993). Self-organisation of genetic coding. *J. Theor. Biol.* 162, 267–287.
- Wills, P. R. (1994). Does information acquire meaning naturally? *Ber. Bunsenges. Phys. Chem.* 98:1129–1134.
- Wills, P. R. (2001). Autocatalysis, Information and Coding. *BioSystems* 60:49–57.
- Wills, P. R. (2009). Informed Generation: physical origin and biological evolution of of genetic codescript interpreters. *J. Theor. Biol.* 257:345–358.
- Markowitz, S., Drummond, A., Nieselt, K. and Wills, P. R. (2006). Simulation model of prebiotic evolution of genetic coding. In Rocha, L. M., Bedau, M., Floreano, D., Goldstone, R., Vespignani, A. and Yeager, L., editors, *Artificial Life X*. pages 152–157. MIT Press, Cambridge, MA.
- Maturana, H. R. and Valera, F. J. (1980). *Autopoiesis and Cognition*. Reidel Publishing, Dordrecht, Holland.

# Evolutionary Transition through a New Multilevel Selection Model

Shelly Xiaonan Wu<sup>1</sup> and Wolfgang Banzhaf<sup>1</sup>

<sup>1</sup>Department of Computer Science, Memorial University, St John's, NL, Canada A1B 5X3  
{xiaonan, banzhaf}@mun.ca

## Abstract

Most multilevel selection models in the literature focus on addressing the evolution of cooperation. There is, however, another aspect of multilevel selection theory. It might be able to provide explanations for evolutionary transitions, which involve the creation of higher level complexes out of simpler elements. Here, we propose a multilevel selection model to support evolutionary transitions. This model employs a genetic operator called “cooperation” to build the hierarchical structure used in multilevel selection theory, and applies two types of multilevel selection to achieve transitions. Our experiments on an extended N-player Prisoner's Dilemma game demonstrate that groups with all required skills emerge from a population of independent individuals, no matter whether skills are equally rewarded or not. Our experiments confirm that both types of multilevel selection mentioned are relevant to evolutionary transitions.

## Introduction

Our biological world is hierarchically organized. Starting from the bottom level to the top, the hierarchy includes atoms, molecules, organelles, cells, tissues, organs, organ systems, organisms, populations, communities, ecosystems and biospheres. It is also generally accepted that the simpler, smaller components appeared before the more complex, composite systems. The creation of new higher level complexes out of simpler entities is referred to as an “evolutionary transition” (Buss, 1987; Michod, 1999; Smith and Szathmáry, 1995).

How and why evolutionary transitions take place during evolution is an important question to address for biologists and sociologists. Increasingly, multilevel selection (MLS) has been suggested as a potent explanation (Michod, 1999; Smith and Szathmáry, 1995; Sober and Wilson, 1999). MLS theory posits that natural selection may simultaneously operate at multiple levels of the biological hierarchy. Multilevel selection theory has its origins in group selection theory, which initially was aimed to explain the evolution of cooperation<sup>1</sup>: Individuals are divided into groups; within-group

selection favors selfish individuals, while between-group selection favors cooperative individuals. When between-group selection dominates within-group selection, a major transition occurs and the group becomes a higher level organism in its own right (Wilson and Wilson, 2007).

The way to explain evolutionary transitions extends MLS theory in an important new way. Nevertheless, investigations of most existing MLS models focus on the conditions necessary for the emergence of cooperation during evolution. The purpose of this paper is to computationally verify the idea that evolutionary transitions can indeed occur through multilevel selection. To this end, we consider a new MLS model and investigate its ability to exploit the division of labor. A crucial step in many of the major transitions (Smith and Szathmáry, 1995) is the division of labor between components of an emerging higher level unit of evolution (Gavrillets, 2010). This new MLS model distinguishes itself from existing MLS models in two ways. First, it integrates two types of multilevel selection (Okasha, 2005), which are believed to be relevant to the evolutionary transitions, each at a different stage. To encourage a transition, group fitness (fitness of higher level units) is defined to be “decoupled” (Michod and Nedelcu, 2003) from the individual fitness (fitness of the lower level units). Second, the model does not take the existence of the hierarchical structure for granted; multicellular organisms do not exist at the beginning of life. Our model constructs the hierarchy through evolutionary transitions. The experiments shown here confirm that in appropriately defined models independent individuals are able to transit to groups with totally different functionalities using multilevel selection; in terms of the division of labor, those are groups with members executing various skills with possibly different rewards.

The remainder of this paper is organized as follows. Section 2 briefly describes multilevel selection theory, especially the two types of multilevel selection. Section 3 in-

ponent of a multilevel theory of evolution. Many strong advocates of other alternatives in explaining the evolution of cooperation have come to accept multilevel analysis (Borrello, 2005; Okasha, 2001, 2008; Wilson, 1983).

<sup>1</sup>Group selection is a longstanding controversial area in the evolution of cooperation. It recently re-emerged as an important com-

troduces our multilevel selection model. Section 4 shows experiments with the model and their results. Section 5 concludes and discusses future work.

## Multilevel Selection

Group selection (Sober and Wilson, 1999) tries to explain the evolution of cooperation by introducing selection between groups. Between-group competition allows traits to arise from evolution that are costly for individuals but beneficial to groups. This is therefore one mechanism by which cooperation is able to emerge in evolution. Individuals and groups, however, are relative: an entity can be regarded as a group for individuals at the level below, and as an individual of a group at the level above. This new perspective is now called multilevel selection (MLS) theory.

When higher level selection (i.e. between-group selection) dominates lower level selection (i.e. within-group selection), an evolutionary transition occurs (Wilson and Wilson, 2007). The reason that individuals would give up their survival and reproductive opportunity to become a part of complexes is that the complexes are able to protect their members from being eliminated by selection. For example, by hunting together or by watching predators for others, members in a group have a greater chance to survive severe competition. In addition, a consequence of higher level selection is adaptation, which minimizes conflict among lower level entities and increases cooperation. Therefore, lower level selection does not interrupt the formation of higher level entities (Okasha, 2005).

For the hierarchical structure used in MLS with a number of individual entities nested within each group entity, we need to clarify which entities should become the objects of evolution or which level should undergo evolution (Okasha, 2005). If we are interested in the changing frequencies of different individual traits, individual entities will be the objects of evolution; group entities are only a structure or an environment where fitness-affecting interactions take place. Most multilevel selection models proposed for the evolution of cooperation, such as Wilson (1975)'s and Traulsen and Nowak (2006)'s models, belong to this kind. These models focus on how to propagate the altruistic trait among individuals in a population. To this end, groups are regularly formed and evaluated. Groups with more altruists will have a higher fitness; hence cooperative individuals in such groups will have higher probabilities to be reproduced. In other words, groups are only temporary fitness-bearing entities; even though they are selected, it is not them but individuals that are reproduced, and also it is the frequency of individual traits that is changed. This type of MLS is called MLS type 1 (MLS1) (Damuth and Heisler, 1988; Okasha, 2005).

Alternatively, if we are interested in the changing frequencies of different group traits, group entities need to be the objects of evolution. They are not merely an environment to in-

dividual entities or an object of selection; they actually have their own heritable traits. Group entities with higher fitness will reproduce more offspring group entities with similar traits. Individual entities may still undergo evolution within each group entity, which leads to changes in the distribution of individual traits and potentially affects group traits. This type of MLS is called MLS type 2 (MLS2) (Damuth and Heisler, 1988; Okasha, 2005). As a result, since the entities undergoing evolution are different in these two types of multilevel selection, the evolutionary changes obtained on each level are different. MLS1 will contribute the most individual entities to the next generation, while MLS2 will contribute the most groups. Both MLS1 and MLS2 are distinct processes that can occur in nature.

According to Okasha (2005), both types of multilevel selection may be relevant to evolutionary transitions. An evolutionary transition is more complicated than the evolution of cooperation. However, before transitions take place and complexes emerge, simpler entities which constitute the complexes have to be able to work together. They need to sacrifice their individuality and exhibit cooperative traits. Therefore, in the early stage of evolutionary transitions, the evolution of cooperation has to emerge, so that cooperative traits can spread among simpler entities in the population. That is exactly what MLS1 promotes: using groups as an environment to help individual traits to propagate. Once individuals are willing to form cohesive complexes, evolution should work on complexes to gradually develop their own traits. In other words, complexes should now themselves become objects of evolution. Through selection and reproduction, complexes are better adapted to their environment and eventually become discrete units, normally with traits different from their constituents' traits. It follows that MLS2 should be applied at a later stage of an evolutionary transition.

The shift from MLS1 to MLS2 also indicates a change in the definition of group fitness. In MLS1, group fitness is defined as the average fitness of the individuals within a group, while in MLS2, group fitness is defined independent of the average fitness of its individuals. As the transition proceeds, group fitness gradually becomes "decoupled" from individual fitness (Michod and Nedelcu, 2003), until it is no longer closely related to the average individual fitness. Once group fitness is decoupled, the transition has been achieved, and new complexes have been created that assume an existence of their own.

## A New MLS Model

The concept of multilevel selection is very simple: levels are like "Russian matryoshka dolls" (Wilson and Wilson, 2008) nested one within another; selection simultaneously operates on every level and favors different types of adaptations. Many models have been proposed based on this concept (see Wu and Banzhaf (2011) for examples). However, their main

focus is to investigate under which conditions the evolution of cooperation will occur or what mechanisms could promote the evolution of cooperation. Furthermore, these models take the hierarchical structure in a MLS for granted; that is, they treat the hierarchical structure as given. Biological hierarchies, on the other hand, have developed gradually; a good example is the evolution of multicellular organisms: it did not exist at the beginning of life. We therefore need to consider other MLS models to explain evolutionary transitions: how simpler entities form complexes and how complexes emerge as discrete units with traits different from their constituents.

This contribution aims at introducing such a new multi-level selection model for evolutionary transitions. The investigation uses the division of labor as an example. Division of labor is a group trait resulting from evolutionary transitions, where low level independent entities with specialized skills cooperate to increase the reproductive success of high level complexes. Examples include the separation of germ and soma cells in simple multicellular organisms, appearance of multiple cell types and organs in more complex organisms, and emergence of casts in eusocial insects (Gavrilets, 2010).

We adopt the extended N-player Prisoner's Dilemma (NPD) game to study the division of labor. The NPD game (Sober and Wilson, 1999) is the classical setting for addressing the evolution of cooperation. Once cooperation is reached, all players possess the same cooperative trait, which is also the only trait required for cooperation. Even if such cooperation breaks down by loosing some individuals, the rest are still capable of cooperating with others. Evidently, the game does not serve the need for investigating the division of labor unless extensions are made. We first change the NPD game by attaching a new trait called "skill" to each player; then we redefine the goal of the NPD game: find  $N$  players who not only are willing to cooperate but also possess all required skills.

The general framework of our model is illustrated in Fig. 1. This model accommodates two types of entities:

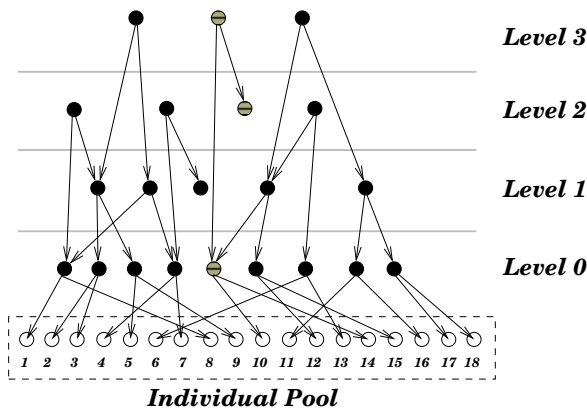


Figure 1: A general framework of the new MLS model

individuals (white circles) and groups (black circles). The initial population contains individuals and groups on level 0, which are composed of two randomly selected individuals. The genome of individuals carries two genes. One gene has two variants (alleles); one allele codes for cooperators, the other for defectors. When the former trait is expressed, the individual is said to be a cooperator; otherwise, it is a defector. The other gene encodes the skill possessed. An individual's fitness is determined by the following equations, depending on whether it is a cooperator (C) or a defector (D):

$$f_{C_i}(x) = base + w \left( \frac{b(n_i q_i - 1)}{n_i - 1} - c \right), \quad (0 \leq i < m) \quad (1)$$

$$f_{D_i}(x) = base + w \frac{b n_i q_i}{n_i - 1}, \quad (0 \leq i < m) \quad (2)$$

where  $m$  is the number of groups in the population;  $base$  the base fitness of cooperators and defectors;  $q_i$  the fraction of cooperators in group  $i$ ;  $n_i$  the size of group  $i$ ;  $b$  and  $c$  are the benefit and cost caused by the altruistic act, respectively;  $w$  is a coefficient. From the above fitness definitions, it becomes clear why the initial population must contain groups on level 0: those groups are the smallest units in which the individual fitness can be evaluated. This fitness definition also implies that cooperation is not supported at the individual level, as cooperators always have lower fitness than defectors. Because individuals are unaware of what skills are needed without higher level entities being formed, the skill trait has no effect on the individual fitness.

Groups in the evolution of cooperation simply pool individuals together; however, groups in our model have their own genotype definition, which is represented by a boolean list. Each position in the list is connected to a unique skill, so that the genotype of a group can keep track of all different skills of its members. When a skill is possessed by at least one cooperator in a group, the corresponding position in the genotype is set to true (we say is activated); when the skill is no longer possessed by any cooperator in that group, we inactivate the position by setting it to false. Again, compared to groups in the evolution of cooperation, groups here require their members to develop different skills, not just to cooperate. As a result, groups exhibit more traits than simply the cooperative trait of individuals. Genetically, groups in our model are ready for evolutionary transitions.

From level 0, an operator called "cooperation" starts to build the hierarchical structure level by level. In each generation, it selects two existing groups proportional to fitness to form a new group. For example, as highlighted in Fig. 1, a group on level 0 and a group on level 2 can be made to cooperate in a new group on level 3. After the cooperation operator is applied, the genotype of the new group contains all unique skills from the two parent groups. This operator allows evolution to tinker with varying group memberships in order to find the best combination of individuals and groups



at lower levels for a higher level function. It in fact is a genetic operator for selecting and reproducing groups; therefore, heritable traits of groups can pass from parent groups to offspring groups. Other genetic operators, such as crossover and mutation, can also be applied to groups. Because groups should be the objects of evolution, multilevel selection of the MLS2 type is employed here.

Group fitness is defined as follows.

$$g(y) = \frac{\sum_{i=0}^n f_{idv}(x_i)}{n} \times \frac{active_{geno}(y)}{length_{geno}(y)} \quad (3)$$

It measures the performance of a group in two respects: (i) the average individual fitness of its  $n$  members and (ii) the percentage of activated skills in the genotype. The intention behind this fitness definition is straightforward; the first part encourages the appearance of cooperators, as cooperators improve the overall individual fitness, and the second part rewards groups in which cooperators possess as many different skills as possible. Obviously, this group fitness is not defined as the average individual fitness, but it can be either proportional to average individual fitness, or completely “decoupled” from individual fitness, depending on the influence of the second term of the fitness function. According to Okasha (2005), the former indicates the transition from MLS1 to MLS2, and the latter indicates the groups have fully emerged as discrete units. Both encourage evolution to reach transitions.

Individuals also evolve. To do so, a group is first selected proportional to fitness; an individual is then selected from this group as a parent. For simplicity, asexual reproduction is considered here. Obviously, even though the survival of individuals is now associated with the performance of their group, individuals at this stage are the objects of evolution. Groups provide context for individual fitness evaluation and selection. Hence, multilevel selection of type MLS1 is applied here.

The specific computational implementation of the framework is shown in Algorithm 1. It begins with initialization.  $N$  individuals,  $r$  percent of which are cooperators, are randomly created and exclusively paired into groups at level 0. Groups at level 0 have their fitness evaluated right away.

In each generation, only one group is created by the cooperation operator, which selects two groups proportional to fitness to create a new group. The consequence of cooperation is the increase of group complexity or the appearance of new levels in the hierarchical structure. To prevent levels from ceaselessly growing, we assign every individual a unique number as its  $ID$ ; no individuals with the same  $ID$  can appear within the same group. After fitness evaluation, the new group is added to the population  $P$ . If at that point the maximum number of groups, say  $N'$ , is reached, another group has to be removed from the population selected inversely proportional to fitness. To highlight the effect of the

---

**Algorithm 1:** Computational Implementation of the New Multilevel Selection Model

---

```

1  $P \leftarrow \text{Initialize\_Population}(N, r);$ 
2  $\text{Evaluate\_Individual\_Fitness}(P);$ 
3  $\text{Evaluate\_Group\_Fitness}(P);$ 
4 while population does not converge or max generation
   is not reached do
5    $gp \leftarrow \text{Conduct\_Cooperation}(P);$ 
6    $\text{Evaluate\_Individual\_Fitness}(gp);$ 
7    $\text{Evaluate\_Group\_Fitness}(gp);$ 
8    $\text{Add\_a\_Group\_to\_Population}(gp, P);$ 
9   if  $\text{Population\_Size}(P) > N'$  then
10     $\text{Remove\_a\_Group}();$ 
11   end
12   for  $i \leftarrow 0$  to  $n$  do
13      $idv \leftarrow \text{Reproduce\_an\_Individual}(P);$ 
14      $\text{Replace\_an\_Individual}(idv, P);$ 
15      $\text{Update\_Changes}(idv, P);$ 
16   end
17 end

```

---

cooperation operator, crossover and mutation on groups are currently not included.

We also asexually reproduce  $n$  individuals every generation. Individuals are selected proportional to fitness from another selected group, instead of from the pool of individuals. The offspring inherits its parent’s genome, and further replaces the genome of a less fit individual in the individual pool. The absolute fitness of individuals in the pool is determined by the average fitness of its copies (i.e. individuals with same  $ID$ ) in all groups. Individuals from the pool are allowed to participate in composing more than one group, so they may have multiple copies in different groups. Depending on group composition, they have different fitness within groups. So the simplest way to determine their absolute fitness is to average the fitness of all copies.

After an individual in the pool is replaced, the change needs to be implemented in all groups that contains the copy of the replaced individual. The group fitness and individual fitness of affected groups need to be updated, accordingly.

We repeat the process until a termination condition has been reached or the population converges.

In summary, this new model distinguishes itself from other multilevel selection models in two ways. First, it integrates two types of multilevel selection, both of which are believed to be relevant to the evolutionary transitions (Okasha, 2005). Individual evolution with the help of group selection is analogous to multilevel selection type 1 (MLS1). It propagates cooperators in the population, which is a prerequisite of evolutionary transitions. Group evolution is then analogous to multilevel selection type 2 (MLS2). The selection pressure on group levels forces groups to evolve adap-

tations for regulating conflicts among their members. The adaptations indicate that groups emerge as discrete entities with heritable traits. Second, instead of taking the hierarchical structure resulting from evolutionary transitions for granted, our model introduces a “cooperation” operator to create higher level complexes out of simpler ones.

## Experiments

In the experiments, we closely examine the transition by our multilevel selection model to the division of labor from a population of independent individuals. First, we examine the ability of our model to evolve groups fulfilling various numbers of skills, when all skills receive the same reward. Second, we examine the dynamics within the model and the responses of individuals when different skills are given different rewards.

### Experimental Setup

The experiments are conducted on the extended NPD game with a population of 200 individuals and a maximum of 50 groups on level 1 and above. The initial fraction of cooperators in the population is 0.5; half of the individuals play cooperators in the game, while the other half are defectors. Eq. 1 and Eq. 2 are used to calculate the fitness of cooperators and defectors within a group, respectively. The base fitness *base* is set to 10, benefit *b* to 5, cost *c* to 1, and coefficient *w* to 1 in these two equations<sup>2</sup>. Group fitness is calculated according to Eq. 3. Group size is a self-adaptive parameter affected by the cooperation operator.

Because the purpose of these experiments is to study the division of labor, our investigation will focus on the effects of two parameters: the number of desired skills and the rewards associated with each skill. For each parameter setting, we ran the model 20 times, each with 5000 generations. We measure the performance of the model by the probability of fixation to cooperators  $P_{fixation}$  and the number of activated skills  $S_{activated}$ .  $P_{fixation}$  is computed as the ratio of the number of runs where population converges to cooperators over 20 runs. We also collect the convergence speed  $S_{converge}$  in each run, which is the number of generations after which group fitness stops to change.

### Varying Skills

The first experiment is given 5 different skills. At initialization, individuals independently choose to be a cooperator or a defector. In addition, they need to randomly pick a skill from 5 skills, {1, 2, 3, 4, 5}. An individual with an attached skill will perform a specific task. The best performing group

should contain only cooperators and should have all 5 skills presented in its genotype. We then gradually increase the number of desired skills to 10, 15 and 20. For each setting, we run the algorithm 20 times. The results are collected in Table 1. The probability of fixation  $P_{fixation}$  with a value of 1 is obtained under all settings, which indicates that defectors, despite a relatively high individual fitness, are eliminated from the population, whereas cooperators dominate the population eventually. MLS1 is the explanation for this result. More importantly, the best performing group for each setting develops all required skills through evolution. This demonstrates that MLS2 is at work. It is not surprising to see the larger the number of desired skills, the slower the population was to reach the equilibrium on group fitness. This is simply a reflection of the problem becoming harder when the number of desired skills is raised.

To get a better idea of how the division of labor develops through evolution, we select a typical run for each of {5, 10, 15, 20} roles for further analysis. Figure 2 depicts the maximum and average number of unique skills of all groups over 500 generations. Starting from at most 2 skills, the best performing groups gradually evolve to perform more and more different skills until the number of desired skills is reached (see Fig. 2a). This growth is due to the guidance provided by the group fitness. Take the run for 20 desired skills for example. We collect the following information from this run: group fitness, the number of activated roles, and the percentage of cooperators in the best performing group, as well as the percentage of cooperators in the population; that is plotted in Fig. 3.

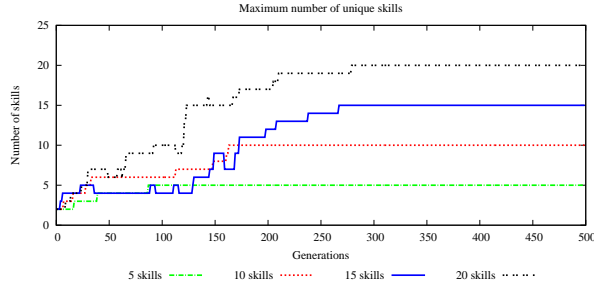
Group fitness (refer to Eq. 3) is determined by the average individual fitness and the percentage of activated skills. We plot the percentage of cooperators, instead of the average individual fitness, in the best group because of two reasons; we can easily extrapolate the average individual fitness from this percentage, and it also shows the fixation process in the best group. Figure 3 clearly shows how the percentage of cooperators and the number of activated roles affect the group fitness. Interestingly, we notice that the population converges to cooperators first, and then the best group develops all required skills. The same trend is also observed in other runs with {5, 10, 15} skills. This observation indicates that cooperators spread in the population before the evolutionary transition happens, a result confirming the discussion about the relationship between MLS1 and MLS2. Group fitness, in turn, influences the execution of individual evolution and group evolution (i.e. cooperation operator). Since defectors bring no fitness benefit on group levels, they are eliminated from the population by group selection at reproduction; hence the percentage of cooperators in the best group and in the population increases steadily towards 1. As shown in Fig. 2b, the average number of activated skills never comes close to the number of desired skills. This implies that the population maintains groups with various

<sup>2</sup>Sensitivity analysis of our model wrt. the initial fraction of cooperators and selection pressure (*w*), as well as a performance comparison with an improved Traulsen’s group selection model (Wu and Banzhaf, 2011) can be found in (Wu, 2011). These experiments confirm that our model promotes cooperation over a wider range of parameter settings.

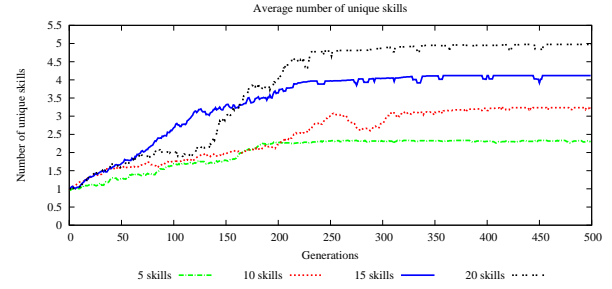


Settings	$P_{fixation}$	$S_{activated}$	$S_{converge}$
$role = 5$	1	5	96.3
$role = 10$	1	10	181.55
$role = 15$	1	15	247.60
$role = 20$	1	20	301.25

Table 1: The performance of our multilevel selection model when individuals play various skills.



(a) Maximum number of unique skills



(b) Average number of unique skills

Figure 2: The changes of the maximum and average number of unique skills in a typical run.

skills. They are potential building blocks, out of which the cooperation operator is able to test different combinations of existing groups, and gradually hones in on optimal groups with all required skills.

In summary, our model is able to successfully evolve groups with all desired skills for the extended NPD game; or we can say that our model is able to evolve groups to engage in the division of labor between equally rewarded skills.

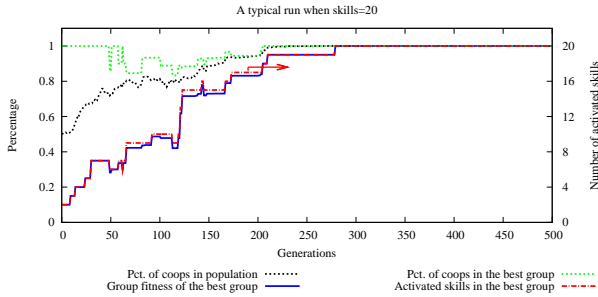


Figure 3: The changes of group fitness, percentage of cooperators and activated roles when 20 skills are set.

## Varying Rewards

We continue the exploration of whether or not our model can evolve the division of labor, but this time skills are unequally rewarded. The different rewards put extra pressure on accomplishing the task, as it attracts individuals to specialize on the most rewarding skills while avoiding the less rewarding skills.

To distinguish skills with different rewards, we refer to the “leader/follower” situation described by Goldsby et al. (2009). Individuals who have skill 1 are appointed as leader of that group, while individuals performing other skills are followers. Leaders receive different reward than followers, but followers, no matter what specific skills they have, receive no other rewards. A coefficient,  $\alpha$ , is used to control how much reward a leader can receive. Coefficient  $\alpha$  basically is a multiplicative of the individual fitness; the individual fitness of a leader is calculated as the product of  $\alpha$  and the individual fitness obtained by Eq. 1 or Eq. 2.

We vary the value of  $\alpha$  in the range of  $\{0.5, 2, 4, 8, 64\}$  on each of  $\{5, 10, 15, 20\}$  roles, and run the model on each setting 20 times. The performance is summarized in Table 2. Clearly for each setting the population converges to cooperators as a result of MLS1, and the best performing group is composed of cooperative individuals with all required skills as a result of MLS2.

Because the group fitness can hardly converge in this experiment, the convergence speed  $S_{converge}$  is judged by the stabilization of  $P_{fixation}$  and  $S_{activated}$ . Fig. 4 displays a typical run when the number of desired skills is set to 5 and coefficient  $\alpha$  is set to 8. Although the percentage of cooperators in the population and the number of activated skills in the best group converge quickly (around generation 350), group fitness and the percentage of leaders in the best group never stop increasing. After generation 350, the percentage of leaders is the only factor that changes the group fitness. Leaders in this case receive much higher rewards than followers, and maximizing this percentage at the same time maximizes the group fitness. Therefore, both values

Settings		$P_{fixation}$	$S_{activated}$	$S_{converge}$
role=5	$\alpha = 0.5$	1	5	90.45
	$\alpha = 2$	1	5	145.35
	$\alpha = 4$	1	5	193.00
	$\alpha = 8$	1	5	238.10
	$\alpha = 64$	1	5	330.00
role=10	$\alpha = 0.5$	1	10	152.2
	$\alpha = 2$	1	10	232.40
	$\alpha = 4$	1	10	379.05
	$\alpha = 8$	1	10	488.00
	$\alpha = 64$	1	10	607.75
role=15	$\alpha = 0.5$	1	15	196.60
	$\alpha = 2$	1	15	313.80
	$\alpha = 4$	1	15	531.50
	$\alpha = 8$	1	15	696.55
	$\alpha = 64$	1	15	950.55
role=20	$\alpha = 0.5$	1	20	314.80
	$\alpha = 2$	1	20	407.35
	$\alpha = 4$	1	20	586.85
	$\alpha = 8$	1	20	902.35
	$\alpha = 64$	1	20	1394.75

Table 2: The performance of runs when leaders are assigned with various rewards.

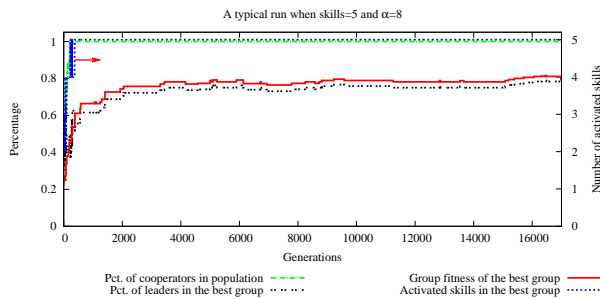


Figure 4: A typical run when skills=5 and  $\alpha=8$ .

are constantly improving. Because there is no upper bound on group size, the cooperation operator keeps creating larger groups with more leaders; therefore an equilibrium distribution of different roles can hardly be reached.

To facilitate the investigation on how different rewards affect the division of labor, we restrict the maximum group size to 20. We plot in Fig. 5 the percentage of leaders in the best performing group collected from a typical run with 5 desired skills when  $\alpha$  is set to each of  $\{0.5, 2, 4, 8, 64\}$ . When  $\alpha$  is set to 0.5, 5% of 20 individuals, which is only 1 individual, play the role as a leader, while when  $\alpha$  is set to 2, 55% of the group, that is 11 individuals, choose to be a leader; similarly, 15 out 20 individuals (75%) become the leader when  $\alpha$  is 4 or 8, and 16 leaders (80%) when  $\alpha$  is 64.

When  $\alpha$  is less than 1, leaders are in fact receiving a penalty, not a reward. Very naturally, individuals avoid be-

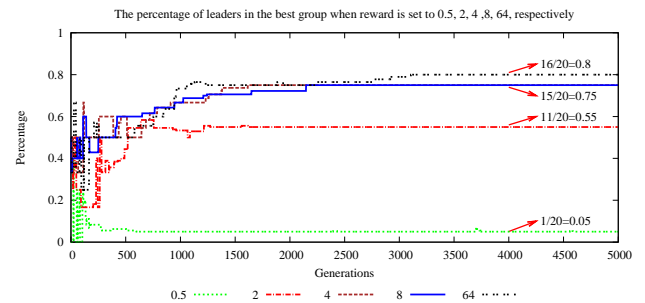


Figure 5: The percentage of leaders in the best group when  $\alpha$  is set to 0.5, 2, 4, 8, 64, respectively.

coming a leader, but because of the selection pressure on the group level, the role of a leader must be present in a group. Therefore, the best group ends up with only 1 leader, which maximizes the group fitness. By contrast, when  $\alpha$  is greater than 1, individuals strive to be leaders because of the positive reward. An  $\alpha$  value of 64 shows another extreme distribution of different roles. Driven by such a significant reward, the best group only has 4 individuals as followers, each for the rest 4 skills, while all other individuals play the role as a leader. The higher the reward, the greater the number of leaders in a group, and the slower the population converges (see  $S_{converge}$  column in Table 2).

The experiment perfectly shows the adaptability of our model in response to changes in group selection pressure, and the importance of selection pressure on group levels in

developing division of labor. Selection pressure eliminates defectors from a population, adjusts the distribution of roles according to the received reward or penalty, and forces all skills to be present even though some of them have lower fitness than others.

## Conclusion

In this paper, we considered a new multilevel selection model to investigate evolutionary transitions. This model introduces a genetic operator called “cooperation” to create higher level complexes out of simpler ones of lower levels. Different types of selection, MLS1 and MLS2, are integrated in the model to determine whether or not the complexes are able to transit to discrete units with their own heritable traits. We test the transition ability of the new model on an extended N-player Prisoner’s Dilemma game for achieving the division of labor from a population of independent individuals. The experiments confirm that our model is able to evolve groups fulfilling various numbers of skills whether skills are equally rewarded or not. The experiments also demonstrate that multilevel selection, both MLS1 and MSL2, are necessary for transitions to occur. MLS1 propagates cooperators in a population. Only when participating individuals are willing to cooperate, will evolutionary transitions occur. MLS2 forces complexes to evolve adaptations for regulating conflicts among their members. The adaptations are guided by group fitness, which in our model is decoupled from individual fitness to promote the appearance of new group traits. In future work, we seek to adapt this model for evolutionary computation to solve problems where transitions are needed.

## Acknowledgments

W.B. would like to acknowledge support from NSERC Discovery Grants, under RGPIN 283304-07. The authors would also like to thank anonymous reviewers for their helpful comments and suggestions.

## References

Borrello, M. E. (2005). The rise, fall and resurrection of group selection. *Endeavour*, 29(1):43–47.

Buss, L. (1987). *The Evolution of Individuality*. Princeton University Press, New Jersey.

Damuth, J. and Heisler, I. L. (1988). Alternative formulations of multilevel selection. *Biology and Philosophy*, 3(4):407–430.

Gavrilets, S. (2010). Rapid transition towards the division of labor via evolution of developmental plasticity. *PLoS Comput Biol*, 6(6):1–10.

Goldsby, H. J., Knoester, D. B., Clune, J., McKinley, P. K., and Ofria, C. (2009). The evolution of division of labor. In *10th European Conference on Artificial Life, Budapest, Hungary, September 2009*.

Michod, R. (1999). *Darwinian Dynamics, Evolutionary Transitions in Fitness and Individuality*. Princeton University Press.

Michod, R. E. and Nedelcu, A. M. (2003). On the reorganization of fitness during evolutionary transitions in individuality. *Integrative and Comparative Biology*, 43(1):64–73.

Okasha, S. (2001). Why won’t the group selection controversy go away? *British Journal for the Philosophy of Science*, 52(2001):25–50.

Okasha, S. (2005). Multi-level selection and the major transitions in evolution. *Philosophy of Science*, 72:1013–1028.

Okasha, S. (2008). Units and levels of selection. In Sarkar, S. and Plutynski, A., editors, *A Companion to the Philosophy of Biology*, chapter 8, pages 138–156. Oxford: Blackwell.

Smith, J. M. and Szathmáry, E. (1995). *The Major Transitions in Evolution*. Oxford University Press.

Sober, E. and Wilson, D. S. (1999). *Unto Others: The Evolution and Psychology of Unselfish Behavior*. Harvard University Press.

Traulsen, A. and Nowak, M. A. (2006). Evolution of cooperation by multilevel selection. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, 103(29):10952–10955.

Wilson, D. S. (1975). A theory of group selection. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, 72(1):143–146.

Wilson, D. S. (1983). The group selection controversy: History and current status. *Annu. Rev. Ecol. Syst.*, 14:159–187.

Wilson, D. S. and Wilson, E. O. (2007). Survival of the selfless. *New Scientist*, 3:42–46.

Wilson, D. S. and Wilson, E. O. (2008). Evolution for the good of the group. *American Scientist*, 96:380–389.

Wu, S. X. (2011). *A Route to the Evolution of Cooperation: Investigations of Multilevel Selection in Evolutionary Computation*. PhD thesis, Department of Computer Science, Memorial University.

Wu, S. X. and Banzhaf, W. (2011). Investigations of Wilson’s and Traulsen’s group selection models in evolutionary computation. In Kampis, G., Karsai, I., and Szathmáry, E., editors, *Proceedings of the 10th European Conference on Artificial Life, Budapest, Hungary, Sept. 13-16, 2009, Part II*, volume 5778 of *Lecture Notes in Computer Science*, pages 1–9. Springer-Verlag Berlin Heidelberg.

# Behavioral Acquisition of Complicated Locomotion for Artificial Elastic Robot using Decentralized Behavior Composed

Keisuke Yoneda<sup>1</sup>, Ikuo Suzuki<sup>1</sup>, Masahito Yamamoto<sup>1</sup> and Masashi Furukawa<sup>1</sup>

<sup>1</sup>Hokkaido University, 9 Nishi, 14 Kita, Kita-ku, Sapporo, 060-0814, Japan  
keisuke-y@complex.ist.hokudai.ac.jp

## Abstract

A virtual elastic robot is proposed which has a body with multiple degrees of freedom. It is capable of fitting its body to the given surrounding environment. This study focuses to allow the elastic robot to adapt to various environments. The intended robot is modeled by rigid objects connected by spring joints in a circular structure. Its control system manipulates spring actuators to realize elastic movements. This paper aims to acquire its control system for the robot to behave autonomously. A behavior acquisition is implemented as an optimization problem by the use of Evolutionary Computation. A physical simulation on the computer is carried out to achieve given tasks for the virtual elastic robot. The task is set to achieve a locomotion which moves toward a destination on a flat ground. Simulation results show that the elastic robot acquires a locomotion. Moreover, we assume a complicated circumstance in which obstacles are placed. In order to allow the robot to adapt to a complicated circumstance, we propose "Behavior Composed" to design a complicated behavior from several simple behaviors. These experimental results prove that the robot is capable of acquiring an adaptive locomotion in specific circumstances.

## Introduction

An autonomous robot is capable of adapting itself to the surrounding environment. This paper focuses on a behavior acquisition for it which can behave to achieve a given task. Accomplishing the task is regarded as a learning problem for autonomous robots. It has been studied actively in areas such as evolutionary robotics and artificial life.

In order to construct an adaptive behavior, a bottom-up approach or evolutionary approach is adopted in recent works. This approach aims to construct an optimum control system so as to achieve a given task by a parameter optimization. It is generally implemented by the use of evolutionary computation. There are many studies to acquire autonomous behaviors by computer simulations (Sims, 1994). As the typical methodology, a virtual creature (Sims, 1994) is proposed to generate the geometric morphology for a model structure and a neural system for controlling a creature automatically. The virtual creature indicated a problem what shape would be optimum to accomplish a task. Similarly, a

behavior emergence and an evolution of the artificial creature become a significant problem in the area of artificial life.

The virtual creature (Sims, 1994) is expected to design the specialized shape in a specific environment. However, the obtained model cannot exploit its adaptive ability in other environment. Therefore, we have focused on an autonomous robot which has a body with multiple degrees of freedom (DOF) and its behavior acquisition. It can behave flexibly such as an amoeba and a snake. There are many studies to design an adaptive behavior for their original robots (Yim et al., 2000; Kamimura et al., 2005; Ishiguro et al., 2008; Yoneda et al., 2009).

The self-reconfigurable robots have a multiple DOF body and consist of simple modules (Yim et al., 2000; Kamimura et al., 2005). Controlling the robot is generally studied by the use of a rule-based control which is described by the specific behavior rule for each module (Yim et al., 2000). Then, an evolutionary heuristic approach is adopted instead of a traditional rule-based approach to control its behavior (Kamimura et al., 2005). However the obtained behaviors are a only simple. The robot cannot behave in the complicated situation. Then, an amoeboid robot (Ishiguro et al., 2008) is proposed to make it possible to behave flexibly. It has a circular structure body which is connected by springs joints and behaves based on a mathematically-modeled protoplasmic streaming motion which is the specific feature of an amoeba. Although it can behave to move toward a light source, it has not obtained composite behaviors.

In those background, we have focused on a circular structure spring robot as a multiple DOF robot, an elastic circular robot (Yoneda et al., 2009). Previously, we have shown that the robot can acquire a locomotion by the use of the decentralized autonomous control system and evolutionary computation. In recent works on behavior acquisition for multiple DOF robots, it is confirmed that the robot can achieve a simple task. For instance, the straight locomotive task on a flat ground without obstacles is regarded as a simple task. However, their robots cannot behave properly in a complicated circumstance in which obstacles are placed. There-

fore, this study aims to acquire an adaptive behavior in a complicated circumstance. We propose "Behavior Composed" which consists of several simple behaviors to design an adaptive behavior for a multiple DOF robot. Learning experiments are carried out to design "Behavior Composed" for the elastic circular robot. Simulation results show the effectiveness of the proposed control approach.

The rest of this paper is composed as follows. Section II explains about a concept of proposed control system, "Behavior Composed". Section III proposes an elastic circular robot consisting of modular units. Section IV describes learning experiments to acquire an autonomous locomotion and shows some experimental results. Section V describes learning experiments to design "Behavior Composed" to adapt to the complicated circumstance. Section VI concludes this study with some remarks and gives some directions toward the future work.

### "Behavior Composed"

In order to acquire an autonomous composite behavior for a mobile robot, the subsumption architecture (Brooks, 1986), which is a layered control system, is proposed as a typical approach. It consists of several primitive behaviors, such as "avoid objects", "wander", "explore", "build maps" and so on. They are assigned to each layer hierarchically based on their priorities. The robot behaves properly by using a selected primitive behavior as the situation demands. This control system shows good performance in a complicated situation. However, its behavior is an unnatural behavior, because each primitive behavior is previously designed.

Then, we have proposed "Behavior Composed" to design an adaptive behavior by the use of Evolutionary Computation (Furukawa et al., 2010). Fig. 1 shows the concept of "Behavior Composed". It consists of several primitive behaviors, "Behavior Simple". "Behavior Simple" is obtained by a learning experiment of a simple task. As the situation demands, the robot combines several "Behavior Simple" to design "Behavior Composed". For instance, a wandering behavior is a combination of three behavior simples, "avoid object", "runaway" and "halt" (Fig. 1).

We have implemented "Behavior Composed" by using an Artificial Neural Network (ANN) (Furukawa et al., 2010). "Behavior Composed" consists of Neural Controller ("Behavior Simple") and Neural Selector (Fig. 2). In the previous work (Yoneda et al., 2010), we conduct two types of learning experiments (simple task and complicated task). The simple task aims to acquire locomotion on a flat ground as a primitive behavior. Then, the complicated task aims to acquire "Behavior Composed" in which obstacle is placed. From experimental results, we confirmed that the robot is capable of an adaptive behavior by switching several "Behavior Simple". This paper focuses on the decentralized "Behavior Composed" that each actuator is controlled by the independently-selected "Behavior Simple".

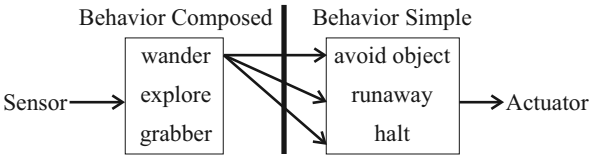


Figure 1: A concept of "Behavior Composed"

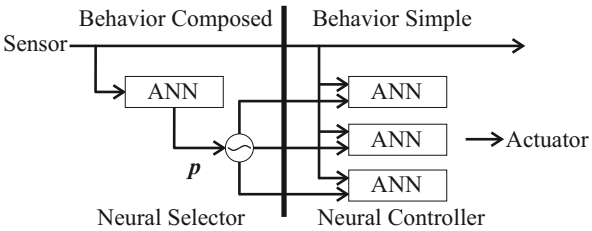


Figure 2: ANN based "Behavior Composed"

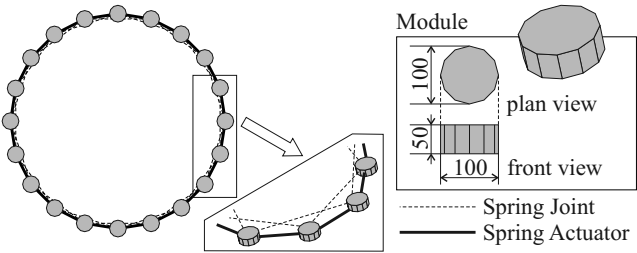


Figure 3: Body system of the elastic robot

### Elastic Circular Robot

#### Body System

An intended elastic robot is modeled by connecting rigid modules circularly with spring joints (Fig. 3). All modules have the same prismatic shape. They are connected to four modules by two spring actuators and two spring joints. This robot can behave by the elastic motion of springs and the friction force between the modules and ground. The robot behavior is controlled by manipulating elastic velocities of each spring actuator. Accordingly, all modules move by propagating spring forces to the whole modules efficiently.

The followings are physical properties used for the robot. The robot is constructed from 20 modules. The density of each module is 2,700[kg/m<sup>3</sup>]. The coefficient of restitution of each module is 0.3. The coefficient of dynamic friction between the modules and the ground is 0.4. The coefficient of static friction between the modules and the ground is 0.6. The natural length of each spring actuator is 0.2[m]. The natural length of each spring joint is 0.4[m]. All spring lengths are able to range from 0.5-fold to 1.5-fold in length. The spring constant is 500[N/m].

#### Sensor System

This elastic robot aims to achieve a locomotive task that the robot mainly moves toward the goal. Then, a goal sensor is installed on each module to perceive a target location. The goal sensor of the *i*-th module is capable of perceiving a

distance  $d_{Li}(t)$  measured from the  $i$ -th module to the goal at time  $t$ . However, when the obstacle is placed between the  $i$ -th module and the goal, the sensor cannot perceive its goal (Fig. 4). Similarly, an obstacle sensor is installed on each module to measure a distance to the nearest obstacle. The obstacle sensor of the  $i$ -th module is capable of perceiving a distance  $d_{Oi}(t)$  which is the shortest distance to the nearest obstacle at time  $t$ .

### Spring Actuator

Expansion and compression of springs make mainly this module move. Each spring actuator is controlled independently by adding an elastic force calculated by Eq. (1).

$$f_i(t + \Delta t) = A_i(t) \sin(\omega_i(t)\Delta t + \phi_i(t)) \quad (1)$$

where  $f_i(t)$  is the elastic force of the  $i$ -th spring actuator at time  $t$ ,  $A_i(t)$  is the amplitude,  $\omega_i(t)$  is the angular velocity,  $\phi_i(t)$  is the accumulated phase ( $\phi_i(0) = 0$ ,  $\phi_i(t + \Delta t) = \phi_i(t) + \omega_i(t)\Delta t$ ). If  $f_i(t) > 0$ , the spring is expanded, and if  $f_i(t) < 0$ , it is compressed. Thus, the behavior of the whole body is controlled by manipulating  $A_i(t)$  and  $\omega_i(t)$  for the  $i$ -th spring actuator.

### Neural Controller

In order to control the robot behavior, each spring actuator has a neural controller. Each neural controller is implemented by ANN to acquire an autonomous behavior in evolution. Neural controller manipulates an elastic force independently for the corresponding spring actuator to realize a decentralized autonomous control system. The controller has eight input neurons and two output neurons to calculate control parameters  $A_i(t)$  and  $\omega_i(t)$ .

In this paper, we implement two types of neural controller. The first one is  $C_L$  which has a light sensor to perceive a light source. The other is  $C_{LO}$  which has a light sensor and an obstacle sensor to perceive a light source and the nearest obstacle. Input parameters of controller are sensor information of connected modules and state variables of the actuator. For two modules  $a$  and  $b$  that are connected by the  $i$ -th spring actuator, the target information ( $L_a(t)$  and  $L_b(t)$ ) is calculated from a distance  $d_{La}(t)$  and  $d_{Lb}(t)$  by Eq. (2). Similarly, The obstacle information ( $O_a(t)$  and  $O_b(t)$ ) is calculated from a distance  $d_{Oa}(t)$  and  $d_{Ob}(t)$  by Eq. (3).

$$L_i(t) = \begin{cases} e^{-\alpha d_{Li}(t)} & (\text{if module } i \text{ receives light}) \\ 0 & (\text{otherwise}) \end{cases} \quad (2)$$

$$O_i(t) = e^{-\beta d_{Oi}(t)} \quad (3)$$

where  $\alpha$  and  $\beta$  are constant values. We set  $\alpha = 0.8$  and  $\beta = 1.0$ .  $L_i(t)$  and  $O_i(t)$  take 1.0 at the maximum value and decays in the inverse ratio to the measured distance.

As state parameters for the  $i$ -th actuator, a current spring length  $l_i(t)$ , an accumulated phase  $\phi_i(t)$ , an amplitude  $A_i(t)$ , phase coherences  $R(\phi_i(t), \phi_{i-1}(t))$ , and

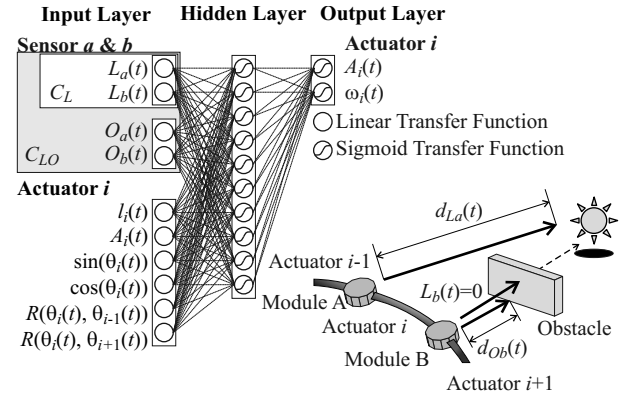


Figure 4: Sensor system and neural controller

$R(\phi_i(t), \phi_{i+1}(t))$  are also input to the controller. The phase coherences are phase differences between the  $i$ -th spring actuator and adjacent actuators. They are calculated by Eq. (4).

$$R(\phi_a(t), \phi_b(t)) = \frac{1}{2} \| e^{j\phi_a(t)} + e^{j\phi_b(t)} \| \quad (4)$$

Then, the controller outputs  $A_i(t)$  and  $\omega_i(t)$  as control parameters for Eq. (1). Fig. 4 shows a configuration diagram of the ANN which has a feed-forward network with a three-layered structure. The controller  $C_L$  has eight input neurons and the controller  $C_{LO}$  has ten input neurons. Both controllers have ten hidden neurons and two output neurons. The synaptic weights of ANN, bias values of neuron and temperature coefficients of the sigmoid function are optimized to acquire an adaptive behavior.

## Acquisition of "Behavior Simple"

### Behavioral Acquisition

A simulation experiment is carried out to achieve a task for a given elastic circular robot. This experiment implements a numerical simulation to optimize parameters of ANN for all spring actuators to acquire an adaptive behavior. All ANNs have the same parameters to simplify the parameter optimization. Then all ANNs are a homogeneous neural controller. In order to optimize parameters assigned to one of ANNs, we adopt the real-coded genetic algorithm (RCGA).

In order to allow the robot to accomplish a task, the simulation environment is implemented by the use of a physics computing library PhysX<sup>1</sup>. PhysX<sup>1</sup> is able to numerically calculate a position and a velocity of the object in consideration of a gravity, a friction, and collision detections. Additionally, we assume a noisy environment as a fluctuation effect in the real world. Then, noises are added to the ANN in each simulation step. The noise is a normal real random number and its strength is 1.0% against each input value.

<sup>1</sup>NVIDIA PhysX

[http://www.nvidia.com/object/physx\\_new.html/](http://www.nvidia.com/object/physx_new.html/)

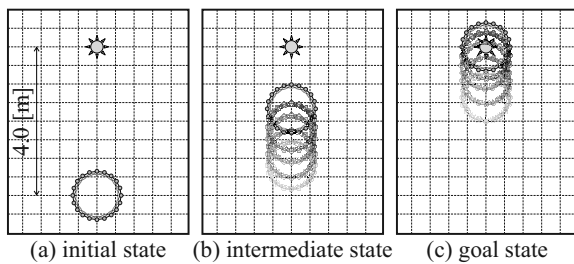


Figure 5: An outline of a task to move toward a light source

## Experimental Conditions

A learning experiment is carried out to acquire an adaptive behavior for the robot. It aims to acquire a locomotive behaviors, moving toward a light source, as a "Behavior Simple". Fig. 5 is an outline of the learning experiment. In this task (Fig. 5), a light source is placed around 4.0[m] from the center of gravity of the robot at the initial condition. The robot behaves autonomously by using the controller  $C_L$ . The obtained behaviors are evaluated by Eq. (5).

$$E_1 = \sum_{t=0}^{N_s} \sum_{i=1}^{N_m} d_{Li}(t) \quad (5)$$

where,  $N_s$  is the number of steps in a simulation and  $N_m$  is the number of modules of which the robot consists.  $E_1$  plays a role of a fitness function to evaluate a photo-tactic behavior. It evaluates an accumulated distances between each module and the light source during one episode simulation. RCGA optimizes the controller  $C_L$  so as to minimize  $E_1$  and results become adaptive behaviors.

As experimental conditions, the simulation step time  $\Delta t$  is 1/60 [sec]. The number of simulation steps in one episode simulation  $N_s$  is 3,600. As optimization conditions for RCGA, the number of individuals is 30, the number of generations is 500, the probability of crossover is 80 [%] and the probability of mutation is 30 [%]. They are determined empirically and this work does not discuss about a difference in those parameters. In order to observe obtained behaviors, we conduct this learning experiment five trials in the same conditions.

## Experimental Results

Fig. 6 shows a diagram which shows the evaluated value along the vertical axis and the number of generations in RCGA along the horizontal axis. The evaluated values for the best and worst trial, and the average evaluated value of all trials are plotted in this diagram. In this figure, the obtained evaluated value of each trial converges as the RCGA generation elapses. Fig. 8 shows snapshots of the obtained behaviors at the 500th generation in RCGA. For this behavior, Fig. 7 shows a diagram which shows the distance between the robot and the light source along the vertical axis and the elapsed simulation steps along the horizontal axis.

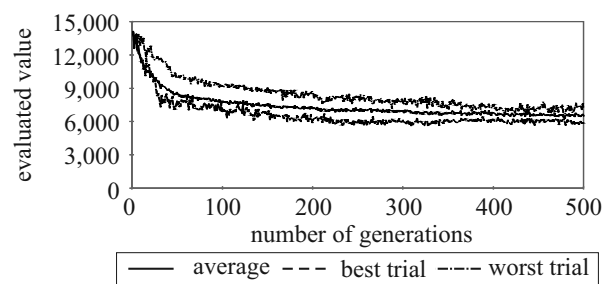


Figure 6: An evaluated value of the locomotive task in each RCGA generation

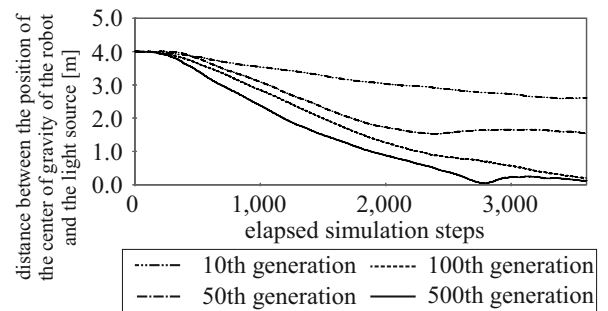


Figure 7: A distance to the destination in each simulation step

As the RCGA generation elapses, it is observed that behaviors are obtained to move toward the light source and stay close to its goal. Therefore, this experiment shows that the robot is capable of achieving the locomotion task.

An optimization experiment shows that the elastic robot acquire a locomotive behavior to achieve a given task. Then, we observe a motion mechanism for the obtained behavior. Fig. 9 shows a diagram which shows the elastic force calculated by using the obtained controller along the vertical axis and the elapsed simulation steps along the horizontal axis. In this charts, lines assigned by "a", "b", "c" and "d" correspond to four actuators respectively in Fig. 8. Fig. 9 indicates a motion at the 1,000th step while the robot is moving. It is observed that four actuators output their elastic forces with the same frequency. Additionally, they make a phase difference locally from a front actuator to a rear actuator in the direction of movement (Fig. 8(b)).

In this way, phase differences of output values make it possible to propagate elastic forces to the whole body effectively. Then, we observe the global phase difference for all actuators. Fig. 10 shows an analysis result for all actuators. The horizontal axis and the vertical axis show the elapsed simulation steps and the label of each actuator. This figure indicates the sign of the elastic force of each actuator in a motion at the 1,000th step. In this chart, the mark "A", "B", "C" and "D" correspond to four parts in Fig. 8. From this result, it is confirmed that all actuators make a phase difference from a front part to a rear part in the movement direction.



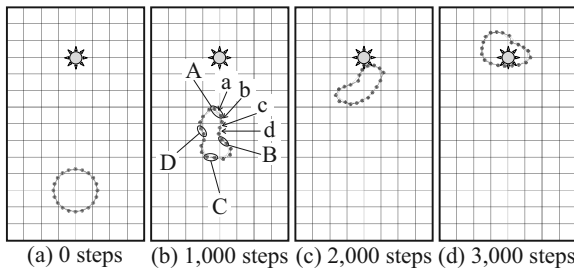


Figure 8: Obtained locomotive behaviors at the 500th generation

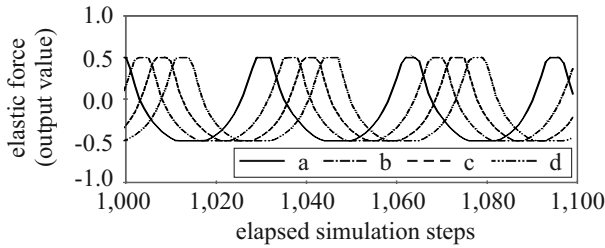


Figure 9: An output elastic force in each simulation step

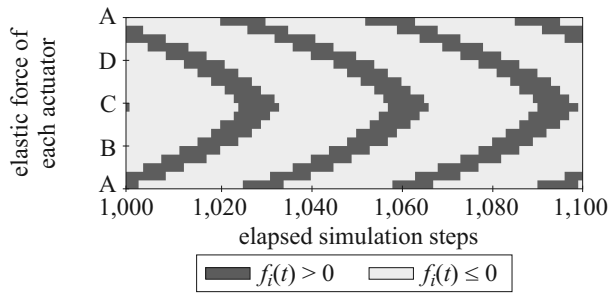


Figure 10: An output sign in each simulation step

## Acquisition of "Behavior Composed"

### Complicated Task

The experiment in the previous section aims to acquire a locomotion which behaves toward a light source. However, the previous experiment is not supposed to accomplish a task in consideration of an obstacle. Fig. 11 is an outline of the complicated task. This task also aims to move toward the light source on a flat ground. However, the obstacle is placed to interrupt light information. In the initial state, the robot can perceive its goal incompletely (Fig. 11(a)). Then, the robot has to explore its goal to reach there.

In order to show the difficulty to achieve the task, we verify behaviors for the locomotive controllers which are obtained in the previous section. Fig. 12, 13, 14 and 15 show the verification results. Fig. 12 and 14 show snapshots of behaviors by using the obtained controller in RCGA trials. Fig. 13 indicates the sign of the elastic force of each actuator in a motion of Fig. 12 at the 1,000th step like Fig. 10. Fig. 15 also indicates the sign of the elastic force of each actuator in a motion of Fig. 14 at the 1,000th step. From Fig. 12 and

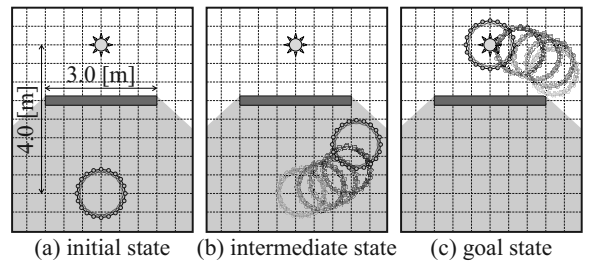


Figure 11: An outline of a task to search a light source

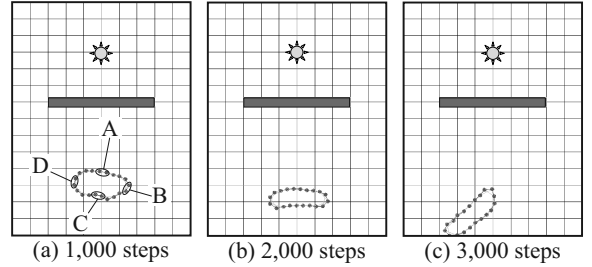


Figure 12: A verification result of the searching task for the locomotive controller obtained in the first trial

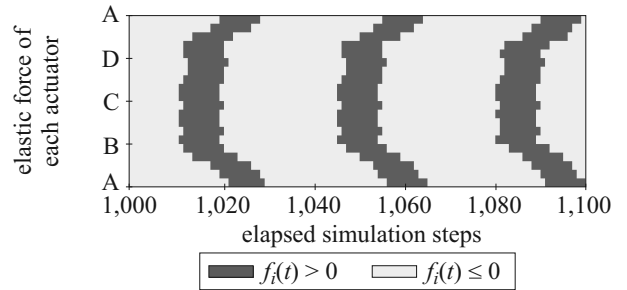


Figure 13: An output sign in each simulation step for the locomotive controller obtained in the first trial

14, those behaviors cannot behave properly. Because, those controllers have not learned the situation which cannot perceive the light source in the previous experiment. Although they cannot achieve the task, we confirm that their behavior has different characteristics in the unlearning situation by comparing Fig. 13 and 15.

### Behavioral Acquisition of "Behavior Composed"

We observed that the robot cannot behave properly by the use of the obtained locomotive controller (Fig. 12 and 14). Then, in the complicated task (Fig. 11), a learning experiment is carried out to acquire an adaptive behavior which avoids the obstacle and moves toward the light source. This experiment aims to acquire "Behavior Composed" to achieve the task. In order to solve this problem, we focus on the obtained locomotive behaviors in the previous section. In the previous experiment, we observed that some behaviors are obtained by conducting optimization trials repeatedly. They are capable of achieving the locomotive task (Fig. 5). However, we confirmed that they have diverse be-

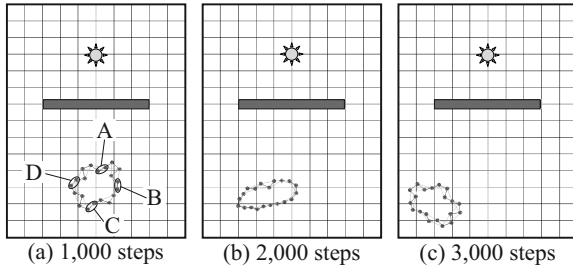


Figure 14: A verification result of the searching task for the locomotive controller obtained in second trial

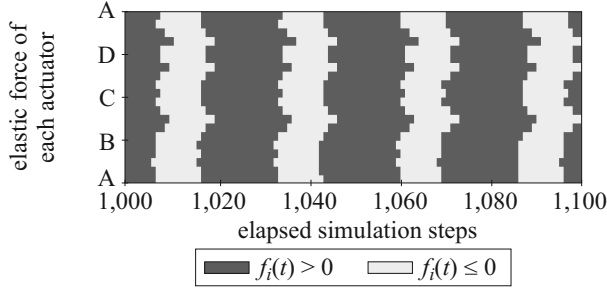


Figure 15: An output sign in each simulation step for the locomotive controller obtained in the second trial

haviors in unlearning situation (Fig. 11). The purpose of this experiment is to examine how to establish "Behavior Composed" by combining diverse behaviors.

### Experimental Conditions

In order to establish "Behavior Composed", a neural selector is installed on each actuator. Fig. 16 shows a configuration diagram of a neural selector. It outputs priorities to determine the appropriate "Behavior Simple" with a maximum priority. Each priority  $p_{ij}$  is corresponding to the  $j$ -th optimized controller ("Behavior Simple") for the  $i$ -th actuator. In this experiment, we set three types of controllers which obtained in the previous section as "Behavior Simple". They are chosen randomly from five obtained controllers, and capable of locomotion as the common capability. Then, each actuator selects an appropriate controller by the use of its neural selector at each simulation step.

This experiment implements numerical simulation to optimize parameters assigned to the ANN for all neural selectors like the previous learning experiment. All ANNs also have the same parameters. The optimization evaluates obtained behaviors based on Eq.(6).

$$E_2 = \sum_{t=0}^{N_s} \sum_{i=1}^{N_m} L_i(t) + A \quad (6)$$

where,  $N_s$  is the number of steps in a simulation,  $N_m$  is the number of modules of which the robot consists, and  $A$  is the transit area cost (Fig. 17). The first term plays a role of a fitness to evaluate a photo-tactic behavior. It evaluates how

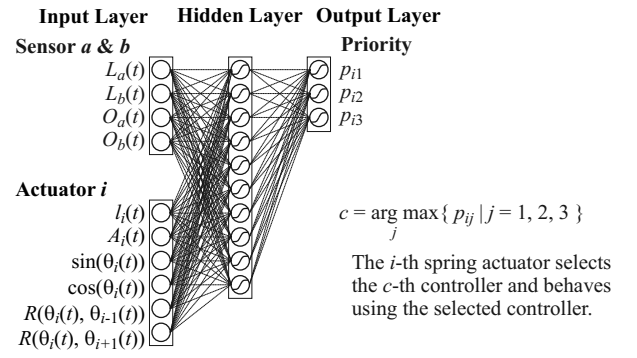


Figure 16: ANN structure of neural selector

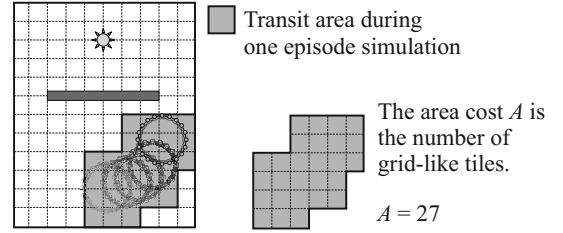


Figure 17: An evaluation of an area cost

much all modules receive a light during one simulation. The second term plays a role of a fitness to evaluate an exploring behavior. Then, when the robot cannot perceive the light source,  $E_2$  evaluates the transit area. RCGA optimizes the neural selector so as to maximize  $E_2$  and results become adaptive behaviors. This optimization is expected to acquire behaviors that a robot avoids the obstacle and moves toward a bright area.

In order to show the effectiveness of the proposed approach, we conduct the same experiment without "Behavior Composed" as a comparison experiment. In this experiment, we set the robot which has the controller  $C_{LO}$  to perceive the light source and the nearest obstacle. This comparison experiment optimizes the controller  $C_{LO}$  like the previous experiment. Then, we confirm the evaluated values and the obtained behaviors for two types of learning experiments.

As experimental conditions, the simulation step time  $\Delta t$  is 1/60 [sec]. The number of simulation steps in one episode simulation  $N_s$  is 5,400. The optimization conditions for RCGA use a set of values the same as the previous experiment. In order to observe obtained behaviors, we also conduct this experiment five trials in the same conditions.

### Experimental Results

Fig. 18, 19 and 20 are experimental results for the optimization of the controller  $C_{LO}$ . Fig. 18 shows a diagram which shows the evaluated value along the vertical axis and the number of generations in RCGA along the horizontal axis. The evaluated values for the best and worst trial, and the average evaluated value of all trials are drawn in this diagrams. For this result, Fig. 19 and 20 show snapshots of the obtained

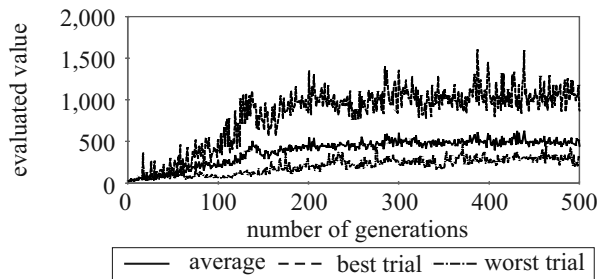


Figure 18: An evaluated value of the searching task in each RCGA generation

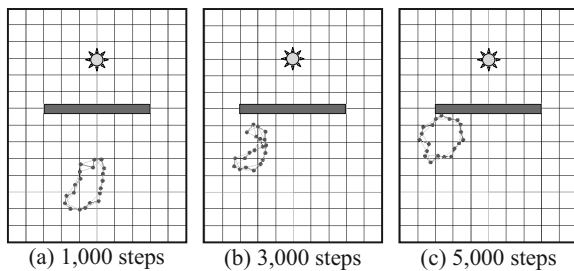


Figure 19: Obtained searching behaviors at the 50th generation

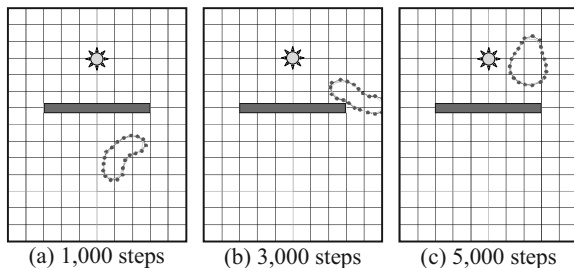


Figure 20: Obtained searching behaviors at the 500th generation

behaviors for the best trial at the 50th and 500th generations in RCGA. It is observed that behaviors are obtained to reach the light source as the RCGA generation elapses. However, The converged evaluated value of the worst trial is about one third of the converged value of the best trial.

Similarly, Fig. 21, 22 and 23 are experimental results for the optimization of a neural selector to acquire "Behavior Composed". Fig. 21 shows a diagram which shows the evaluated value along the vertical axis and the number of generations in RCGA along the horizontal axis. Fig. 22 and 23 show snapshots of the obtained behaviors for the best trial at the 50th and 500th generations in RCGA. In this result, it is observed that the obtained behavior achieves the given task even a behavior obtained at the 50th generation (Fig. 22). Moreover, each trial obtains a better evaluated value than one of the comparison experiment. Then, these results show that "Behavior Composed" has a good performance to establish an adaptive behavior.

For the obtained behavior (Fig. 23), we observe a feature

of the optimized neural selector. Fig. 24 indicates the elastic force of each actuator at the 1,000th step. The horizontal axis and the vertical axis show the elapsed simulation steps and the label of each actuator. Fig. 25 shows a selected "Behavior Simple" in a motion at the 1,000 step. In those figures, the mark "A", "B", "C" and "D" correspond to four parts in Fig. 23(a). From Fig. 24, the robot can behave making proper phase difference like Fig. 10. Additionally, from Fig. 25, we observe that the obtained neural selector switches two "Behavior Simple" periodically. In particular, by comparing Fig. 24 and 25, we also confirm that the switching frequency of "Behavior Simple" equals to the frequency of an elastic motion. Then, the neural selector would switch "Behavior Simple" to make the specific rhythm pattern and phase differences between neighboring actuators.

Now, we observe the behavior, which reaches the light source, in a motion. Fig. 26 shows a selected "Behavior Simple" in a motion at the 3,000th step. Then, when the robot reaches the light source, its neural selector mainly uses one "Behavior Simple". This mean that each "Behavior Simple" has a locomotive ability in this situation.

## Conclusions and Future Works

We have focused on an elastic robot and its physical simulation. Behavioral acquisition for the virtual elastic robot in simulation can be regarded as a learning problem how the robot acquires the adaptive behavior. Evolutionary computation is a successful approach to this learning problem. This study is summarized as follows.

1. A locomotive behavior is acquired when each actuator makes a phase difference of an elastic force from a front actuator to a rear one in the movement direction.
2. "Behavior Composed" is acquired to adapt to the unlearning situation by combining obtained locomotive behaviors, "Behavior Simple".

The searching task shows that the proposed approach is capable of acquire an adaptive behavior at the only 50th generation. It is equivalent to the obtained behavior which obtained at the 500th generation without "Behavior Composed". Then, our approach is possible to acquire proper behavior effectively in a short time. For these experiments, moving images are put on our website<sup>2</sup>. The followings are the rest as some challenges in a future work.

1. Analyzing a switching mechanism mathematically for the obtained "Behavior Composed".
2. Applying "Behavior Composed" to the other multiple DOF robots which have other types of body structure.

<sup>2</sup>Autonomous System Engineering Lab., Hokkaido Univ.  
<http://autonomous.complex.eng.hokudai.ac.jp/researches/physics-modeling/movies/yoneda/>

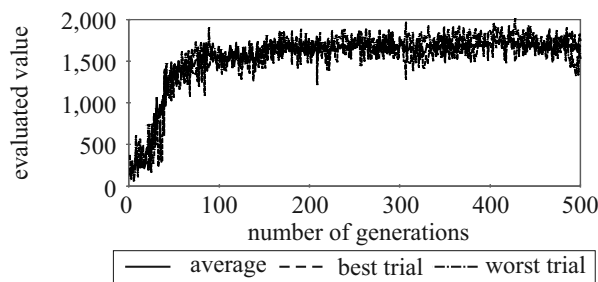


Figure 21: An evaluated value of the searching task for "Behavior Composed" in each RCGA generation

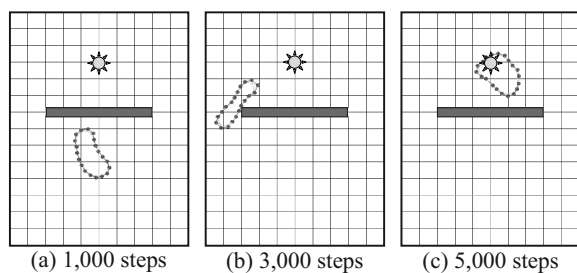


Figure 22: Obtained "Behavior Composed" for the searching task at the 50th generation

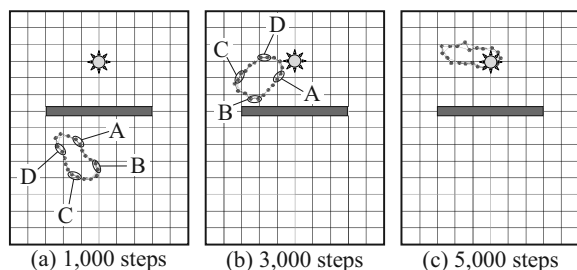


Figure 23: Obtained "Behavior Composed" for the searching task at the 500th generation

## References

- Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23.
- Furukawa, M., Morinaga, M., Ooe, R., Watanabe, M., Suzuki, I., and Yamamoto, M. (2010). Behavior composed for artificial flying creature. In *Joint 5th International Conference on Soft Computing and Intelligent Systems and 11th International Symposium on Advanced Intelligent systems*.
- Ishiguro, A., Umedachi, T., Kitamura, T., Nakagaki, T., and Kobayashi, R. (2008). A fully decentralized morphology control of an amoeboid robot by exploiting the law of conservation of protoplasmic mass. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- Kamimura, A., Kurokawa, H., Yoshida, E., Murata, S., Tomita, K., and Kokaji, S. (2005). Automatic locomotion design and experiments for a modular robotic system. *IEEE/ASME Trans. Mechatronics*, 10(3):314–325.
- Sims, K. (1994). Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques, SIGGRAPH '94*, pages 15–22, New York, NY, USA. ACM.
- Yim, M., Duff, D. G., and Roufas, K. D. (2000). Polybot: a modular reconfigurable robot. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, volume 1, pages 514–520.
- Yoneda, K., Suzuki, I., Yamamoto, M., and Furukawa, M. (2009). Acquisition of adaptive behavior for virtual modular robot using evolutionary computation. In *The 10th European Conference on Artificial Life*.
- Yoneda, K., Suzuki, I., Yamamoto, M., and Furukawa, M. (2010). Acquisition of locomotion for elastic circular robot by using composite behavior. In *Joint 5th International Conference on Soft Computing and Intelligent Systems and 11th International Symposium on Advanced Intelligent systems*.

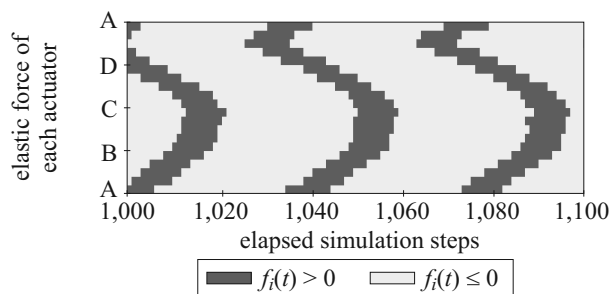


Figure 24: An output elastic force in each simulation step for the obtained "Behavior Composed"

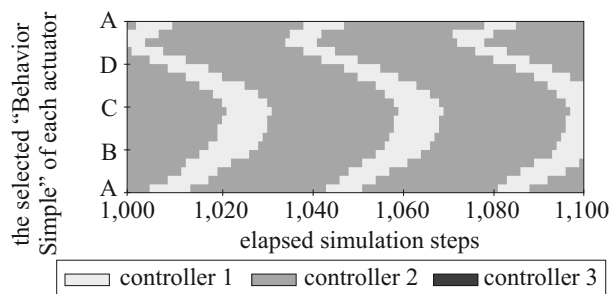


Figure 25: A selected "Behavior Simple" in each simulation step in reaching the obstacle

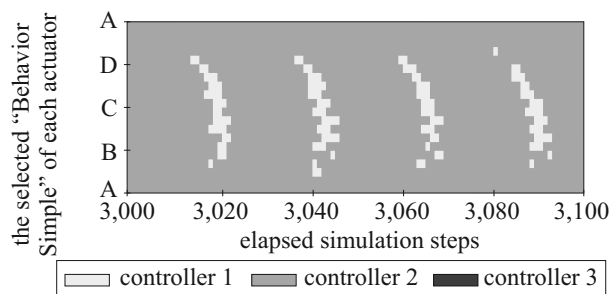


Figure 26: A selected "Behavior Simple" in each simulation step in moving toward the light source

# Evolving Robot Gaits in Hardware: the HyperNEAT Generative Encoding Vs. Parameter Optimization

Jason Yosinski<sup>1</sup>, Jeff Clune<sup>1</sup>, Diana Hidalgo<sup>1</sup>, Sarah Nguyen<sup>1</sup>, Juan Cristobal Zagal<sup>2</sup>, and Hod Lipson<sup>1</sup>

<sup>1</sup> Cornell University, 239 Upson Hall, Ithaca, NY 14853, USA

<sup>2</sup> University of Chile, Beauchef 850, Santiago 8370448, Chile  
yosinski@cs.cornell.edu

## Abstract

Creating gaits for legged robots is an important task to enable robots to access rugged terrain, yet designing such gaits by hand is a challenging and time-consuming process. In this paper we investigate various algorithms for automating the creation of quadruped gaits. Because many robots do not have accurate simulators, we test gait-learning algorithms entirely on a physical robot. We compare the performance of two classes of gait-learning algorithms: locally searching parameterized motion models and evolving artificial neural networks with the HyperNEAT generative encoding. Specifically, we test six different parameterized learning strategies: uniform and Gaussian random hill climbing, policy gradient reinforcement learning, Nelder-Mead simplex, a random baseline, and a new method that builds a model of the fitness landscape with linear regression to guide further exploration. While all parameter search methods outperform a manually-designed gait, only the linear regression and Nelder-Mead simplex strategies outperform a random baseline strategy. Gaits evolved with HyperNEAT perform considerably better than all parameterized local search methods and produce gaits nearly 9 times faster than a hand-designed gait. The best HyperNEAT gaits exhibit complex motion patterns that contain multiple frequencies, yet are regular in that the leg movements are coordinated.

## Introduction and Background

Legged robots have the potential to access many types of terrain unsuitable for wheeled robots, but doing so requires the creation of a gait specifying how the robot walks. Such gaits may be designed either manually by an expert or via computer learning algorithms. It is advantageous to automatically learn gaits because doing so can save valuable engineering time and allows gaits to be customized to the idiosyncrasies of different robots. Additionally, learned gaits have outperformed engineered gaits in some cases (Hornby et al., 2005; Valsalam and Miikkulainen, 2008).

In this paper we compare the performance of two different methods of learning gaits: parameterized gaits optimized with six different learning methods, and gaits generated by evolving neural networks with the HyperNEAT generative encoding (Stanley et al., 2009). While some of these

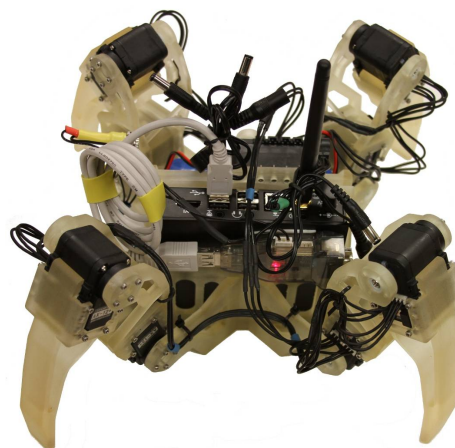


Figure 1: The quadruped robot for which gaits were evolved. The translucent parts were produced by a 3D printer. Videos of the gaits can be viewed at <http://bit.ly/ecalgait>

methods, such as HyperNEAT, have been tested in simulation (Clune et al., 2009a, 2011), we investigate how they perform when evolving on a physical robot (Figure 1).

Previous work has shown that quadruped gaits perform better when they are *regular* (i.e. when the legs are coordinated) (Clune et al., 2009a, 2011; Valsalam and Miikkulainen, 2008). For example, HyperNEAT produced fast, natural gaits in part because its bias towards regular gaits created coordinated movements that outperformed gaits evolved by an encoding not biased towards regularity (Clune et al., 2009a, 2011). One of the motivations of this paper is to investigate whether any learning method biased towards regularity would perform well at producing quadruped gaits, or whether HyperNEAT's high performance is due to additional factors, such as its abstraction of biological development (described below). We test this hypothesis by comparing HyperNEAT to six local search algorithms with a parametrization biased toward regularity.

An additional motivation is to test whether techniques for evolving gaits in simulation, especially cutting-edge evolu-

tionary algorithms, transfer to reality well. Because HyperNEAT gaits performed well in simulation, it is interesting to test whether HyperNEAT can produce fast gaits for a physical robot, including handling the noisy, unforgiving nature of the real world. Such tests help us better understand the real world implications of results reported only in simulation. It is additionally interesting to test how more traditional gait optimization techniques compete with evolutionary algorithms when evolving in hardware. A final motivation of this research is simply to evolve effective gaits for a physical robot.

## Related Work

Various machine learning techniques have proved to be effective at generating gaits for legged robots. Kohl and Stone presented a policy gradient reinforcement learning approach for generating a fast walk on legged robots (Kohl and Stone, 2004), which we implemented for comparison. Others have evolved gaits for legged robots, producing competitive results (Chernova and Veloso, 2005; Hornby et al., 2005; Zykov et al., 2004; Clune et al., 2009a, 2011, 2009b,c; Téllez et al., 2006; Valsalam and Miikkulainen, 2008). In fact, an evolved gait was used in the first commercially-available version of Sony’s AIBO robot (Hornby et al., 2005). Except for work with HyperNEAT (Clune et al., 2009a, 2011, 2009b,c), the previous evolutionary approaches have helped evolution exploit the regularity of the problem by manually decomposing the task. Experimenters have to choose which legs should be coordinated, or otherwise facilitate the coordination of motion. Part of the motivation of this paper is to compare the regularities produced by HyperNEAT to those generated by a more systematic exploration of regularities via a parameterized model.

## Problem Definition

The gait learning problem aims to find a *gait* that maximizes some performance metric. Mathematically, we define a gait as a function that specifies a vector of commanded motor positions for a robot over time. We can write gaits without feedback — also called open-loop gaits — as

$$\vec{x} = g(t) \quad (1)$$

for commanded position vector  $\vec{x}$ . The function depends only on time.

It follows that open-loop gaits are deterministic, producing the same command pattern each time they are run. While the commanded positions will be the same from trial to trial, the actual robot motion and measured fitness will vary due to the noisiness of trials in the real world.

For the system evaluated in this paper, we chose to compare open-loop gaits generated by both the parameterized methods and HyperNEAT. An interesting extension would

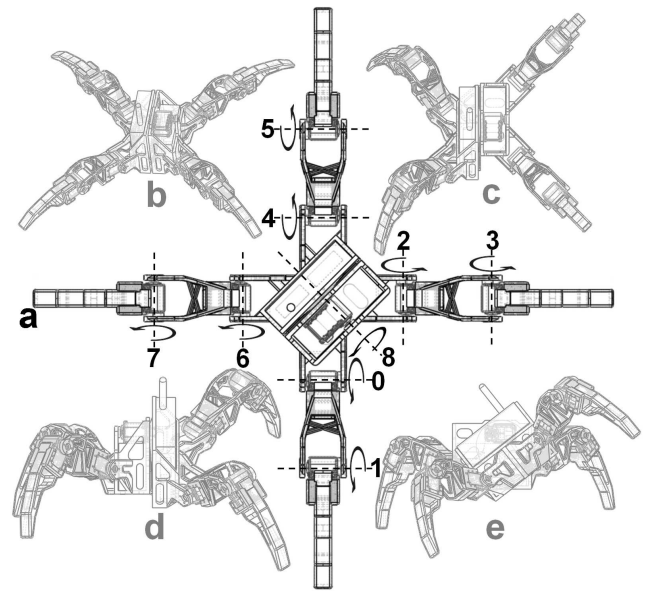


Figure 2: (a) Top-down perspective of the robot with the nine joints and associated servos labeled. (b) The robot in a flat pose with the hip joint centered. (c,d,e) Various views of a pose in which the hip joint is rotated.

be to allow closed-loop gaits that depend on the measured servo positions, loads, voltage drops, or other quantities.

The ultimate goal was to design gaits that were as fast as possible. Our performance metric was thus displacement over the evaluation period of 12 seconds. Details of how this displacement was measured are given below.

## Experimental Setup

### Platform Details

The quadruped robot in this study was assembled from off-the-shelf components and parts printed on the Objet Connex 500 3-D Printing System. It weighs 1.88 kg with the on-board computer and measures approximately 38 centimeters from leg to opposite leg in the crouch position depicted in Figure 1. The robot is actuated by 9 AX-12+ Dynamixel servos: one inner joint and one outer joint servo in each of the four legs, and one servo at the center “hip” joint. This final unique servo allows the two halves of the robot to rotate with respect to each other. Figure 2 shows this unique motion, as well as the positions and numerical designations of all nine servos. Each servo could be commanded to a position in the range  $[0, 1023]$ , corresponding to a physical range  $[-120^\circ, +120^\circ]$ . The computer and servos can be powered by two on-board batteries, but for the tests presented in this paper power was provided by a tethered cable.

All of the computation for gait learning, fitness evaluation, and robot control was performed on the compact, on-board CompuLab Fit-PC2, running Ubuntu Linux 10.10.





Figure 3: A Nintendo Wii remote provided the location of the robot by tracking the infrared LED mounted on the robot’s antenna. The position was measured in pixels and transmitted from the Wii remote to the robot via bluetooth.

The slowest portion of code was HyperNEAT, which took less than one second per generation to run (excluding physical evaluations). Thus, we chose not to offload any computation. All gait generation, learning, and fitness evaluation code, except HyperNEAT, was written in Python and is available on our website (<http://bit.ly/ecalgait>). HyperNEAT is written in C++. We controlled the servos with the Pydynamixel library, sending commanded positions at 40Hz. The robot connected to a wireless network on boot, which enabled us to control it via SSH.

Robot gaits are defined by a Python *gait function* that takes time (starting at 0) as a single input and outputs a list of nine commanded positions (one for each servo). To safeguard against limb collision with the robot body, the control code cropped the commands to a safe range. This range was  $[-85^\circ, +60^\circ]$  for the inner leg servos,  $[-113^\circ, +39^\circ]$  for the outer leg servos, and  $[-28^\circ, +28^\circ]$  for the center hip servo.

### Fitness Evaluation Details

To track the position of the robot and thus determine gait fitness, we mounted a Nintendo Wii remote on the ceiling and an infrared LED on top of the robot (Figure 3). The Wii remote contains an IR camera that tracks and reports the position of IR sources. The resolution of the camera was 1024 by 768 pixels with view angles of about  $40^\circ$  by  $30^\circ$ , which produced a resolution of 1.7mm per pixel when mounted at a height of 2.63m. At this height, the viewable window on the floor was approximately 175 x 120 cm.

A separate Python tracking server ran on the robot and interfaced with the Wii remote via bluetooth using the CWiid library. Our fitness-testing code communicated with this server via a socket connection and requested position updates at the beginning and end of each run.

As mentioned earlier, the metric for evaluating gaits was the Euclidian distance the robot moved during a 12-second run on flat terrain. For the manual and parameterized gaits, the fitness was this value. The HyperNEAT gaits stressed the motors more than the other gaits, so to encourage gaits that did not tax the motors we penalized gaits that caused the servos to stop responding. When the servos stopped responding they could, in nearly all cases, be restarted by cycling power, though over the course of this study we did have to replace four servos that were damaged. The penalty was to set the fitness to half of the distance the robot actually traveled. We tested whether the servos were responding after each gait by commanding them to specific positions and checking whether they actually moved to those positions. This test had the additional benefit of rewarding those gaits that did not flip the robot into a position where it could not move its legs, which HyperNEAT also did more than the other learning methods. Because the fitness of HyperNEAT gaits were often halved, in results we compare actual distance traveled in addition to fitness for the best gaits produced by each class of gait-generating algorithms.

Since only a single point on the robot — the IR LED — was measured for the purposes of computing fitness, it was important that the position of the IR LED accurately reflect the position of the robot as a whole. To enforce this constraint, the robot was always measured while in the *ready* position (the position shown in Figure 1). This was done to prevent assigning extra fitness to, for example, gaits that ended with the robot leaning toward the direction of travel (this extra distance would not likely generalize in a longer run, which is why we did not want to reward this behavior).

In order to measure the start and end position in the same pose, and to ensure fair fitness evaluations with as little noise as possible, we linearly interpolated the motion of the robot between the ready position and the commanded gait,  $g(t)$ . As shown in Figure 4, the instantaneous robot limb configuration during the first and last portions of the evaluation was an interpolation between the initial ready position and  $g(t)$ ; during the rest of the evaluation, the robot followed the commanded gait exactly.

The only human intervention required during most learning trials was to occasionally move the robot back into the viewable area of the Wii remote whenever it left this window. Initially this was a rare occurrence, as the gaits did not typically produce motion as large as the size of the window (roughly 175 x 120 cm). However, as gaits improved, particularly when using HyperNEAT, the robot began to walk out of the measurement area a non-negligible fraction of the time. Whenever it did so, we would discard the trial and



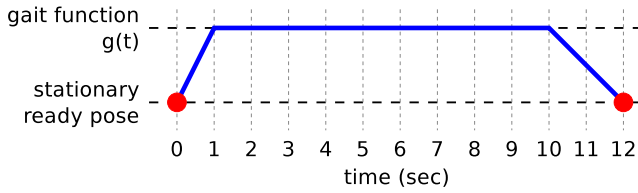


Figure 4: Motion was interpolated linearly between a stationary pose and the commanded gait  $g(t)$  for one second at the beginning of each run and two seconds at the end, as shown above. The position of the robot was measured at the beginning and end of each run (red circles) in the ready pose.

repeat it until the gait finished within the window. While this process guaranteed that we always obtained a measurement for a given gait before proceeding, it also biased some measurements downward. Because the performance of the robot on a given gait varied from trial to trial, a successful measurement was more likely to be obtained when the gait happened to perform poorly. This phenomenon was negligible at first, but became more pronounced as gaits began traversing the entire area. HyperNEAT gaits were especially likely to require additional trials, meaning that the reported performance for HyperNEAT is worse than it would have been otherwise. Future studies could employ an array of Wii remotes to increase the size of the measurement arena.

## Gait Generation and Learning

We now describe the classes of gait-generating algorithms.

### Parameterized Gaits

By a *parameterized gait*, we mean a gait produced by a parameterized function  $g(t; \vec{\theta})$ . Fixing the parameters  $\vec{\theta}$  yields a deterministic motion function over time. We tried several parametrizations on the robot and, upon obtaining reasonable early success, settled on one particular parametrization, which we call *SineModel5*. Its root pattern is a sine wave and it has five parameters (Table 1).

Intuitively, *SineModel5* starts with 8 identical sine waves of amplitude  $\alpha$  and period  $\tau$ , multiplies the waves for all outer motors by  $m_O$ , multiplies the waves for all front motors by  $m_F$ , and multiplies the waves for all right motors by

Parameters in $\vec{\theta}$	Description	Range
$\alpha$	Amplitude	[0, 400]
$\tau$	Period	[.5, 8]
$m_O$	Outer-motor multiplier	[-2, 2]
$m_F$	Front-motor multiplier	[-1, 1]
$m_R$	Right-motor multiplier	[-1, 1]

Table 1: The *SineModel5* motion model parameters.

$m_R$ . To obtain the actual motor position commands, these waves are offset by fixed constants ( $C_O = 40$  for outer motors,  $C_I = 800$  for inner motors, and  $C_C = 512$  for the center hip motor) so that the base position (when the sine waves are at 0) is approximately a crouch (the position shown in Figure 1). To keep the size of the model search space as small as possible, we decided to keep the ninth (center) motor at a fixed neutral position. Thus, the commanded position for each motor as a vector function of time is as follows (numbered as in Figure 2):

$$\vec{g}(t) = \begin{bmatrix} \alpha \cdot \sin(2\pi t/\tau) \cdot m_F + C_I \\ \alpha \cdot \sin(2\pi t/\tau) \cdot m_O \cdot m_F + C_O \\ \alpha \cdot \sin(2\pi t/\tau) + C_I \\ \alpha \cdot \sin(2\pi t/\tau) \cdot m_O + C_O \\ \alpha \cdot \sin(2\pi t/\tau) \cdot m_R + C_I \\ \alpha \cdot \sin(2\pi t/\tau) \cdot m_O \cdot m_R + C_O \\ \alpha \cdot \sin(2\pi t/\tau) \cdot m_F \cdot m_R + C_I \\ \alpha \cdot \sin(2\pi t/\tau) \cdot m_O \cdot m_F \cdot m_R + C_O \\ 0 + C_C \end{bmatrix}$$

### Learning Methods for Parameterized Gaits

Given the *SineModel5* parameterized motion model (see previous section) and the allowable ranges for its five parameters (Table 1), the task is discovering values for the five parameters that result in fast gaits.

If we choose a value for the five dimensional parameter  $\vec{\theta}$ , then a given physical trial gives us one measurement of the fitness  $f(\vec{\theta})$  of that parameter vector. Two things make learning difficult. First, each evaluation of  $f(\vec{\theta})$  is expensive, taking 15-20 seconds on average. Second, the fitness returned by such evaluations has proved to be very noisy, with the standard deviation of the noise often being roughly equivalent to the size of the measurement.

We test the ability of different *learning algorithms* to choose the next value of  $\vec{\theta}$  to try, given a list of the  $\vec{\theta}$  values already evaluated and their fitness measurements  $f(\vec{\theta})$ .

We evaluated the following six different learning algorithms for the parameterized motion models:

*Random*: This method randomly generates parameter vectors in the allowable range for every trial. This strategy serves as a baseline for comparison.

*Uniform random hill climbing*: This method repeatedly starts with the current best gait and then selects the next  $\vec{\theta}$  by randomly choosing one parameter to adjust and replacing it with a new value chosen with uniform probability in the allowable range for that parameter. This new point is evaluated, and if it results in a longer distance walked than the previous best gait, it is saved as the new best gait.

*Gaussian random hill climbing*: This method works similarly to Uniform random hill climbing, except the next  $\vec{\theta}$  is generated by adding random Gaussian noise to the current best gait. This results in all parameters being changed at once, but the resulting vector is always fairly close to the

previous best gait. We used independently selected noise in each dimension, scaled such that the standard deviation of the noise was 5% of the range of that dimension.

*N-dimensional policy gradient ascent:* We implemented Kohl and Stone's (Kohl and Stone, 2004) method for local gradient ascent for gait learning with noisy fitness evaluations. This strategy explicitly estimates the gradient of the objective function. It does this by first generating  $n$  parameter vectors near the initial vector by perturbing each dimension of each vector randomly by either  $-\epsilon$ , 0, or  $\epsilon$ . Then each vector is run on the robot, and for each dimension we segment the results into three groups:  $-\epsilon$ , 0, and  $\epsilon$ . The gradient along this dimension is then estimated as the average score for the  $\epsilon$  group minus the average score for the  $-\epsilon$  group. Finally, the method creates the next  $\vec{\theta}$  by changing all parameters by a fixed-size step in the direction of the gradient. For this study we used values of  $\epsilon$  equal to 5% of the allowable range in each dimension (ranges listed in Table 1), and a step size scaled such that if all dimensions were in the range  $[0, 1]$ , the norm of the step size would be 0.1.

*Nelder-Mead simplex method:* The Nelder-Mead simplex method creates an initial simplex with  $d + 1$  vertices for a  $d$  dimensional parameter space. It then tests the fitness of each vertex and, in general, it reflects the worst point over the simplex's centroid in an attempt to improve it. Several additional rules are used to prevent cycles and local minima; see Singer and Nelder (2009) for more information.

*Linear regression:* To initialize, this method chooses and evaluates five random parameter vectors. It then fits a linear model from parameter vector to fitness. In a loop, the method chooses and evaluates a new parameter vector generated by taking a fixed-size step in the direction of the gradient for each parameter, and fits a new linear model to all vectors evaluated so far, choosing the model to minimize the sum of squared errors. The step size is the same as in *N-dimensional policy gradient ascent*.

Three runs were performed per learning method. To most directly compare learning methods, we evaluated the different methods by starting each of their three runs, respectively, with the same three randomly-chosen initial parameter vectors ( $\vec{\theta}_A$ ,  $\vec{\theta}_B$ , and  $\vec{\theta}_C$ ). Runs continued until the performance plateaued, which we defined as when there was no improvement during the last third of a run.

## HyperNEAT Gait Generation and Learning

HyperNEAT is an indirect encoding for evolving artificial neural networks (ANNs) that is inspired by the way natural organisms develop (Stanley et al., 2009). It evolves Compositional Pattern Producing Networks (CPPNs) (Stanley, 2007), each of which is a genome that encodes an ANN phenotype (Stanley et al., 2009). Each CPPN is itself a directed graph, where the nodes in the graph are mathematical functions, such as sine or Gaussian. The nature of these functions can facilitate the evolution of properties such as sym-

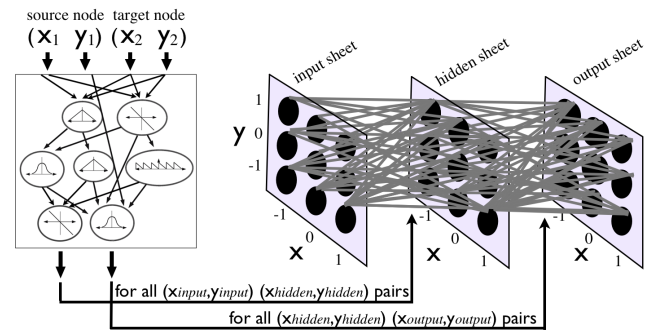


Figure 5: HyperNEAT produces ANNs from CPPNs. ANN weights are specified as a function of the geometric coordinates of each connection's source and target nodes. These coordinates and a constant bias are iteratively passed to the CPPN to determine each connection weight. The CPPN has two output values, which specify the weights for each connection layer as shown. Figure from Clune et al. (2011).

metry (e.g. a Gaussian function) and repetition (e.g. a sine function) (Stanley et al., 2009; Stanley, 2007). The signal on each link in the CPPN is multiplied by that link's weight, which can magnify or diminish its effect.

A CPPN is queried once for each link in the ANN phenotype to determine that link's weight (Figure 5). The inputs to the CPPN are the Cartesian coordinates of both the source (e.g.  $x = 2$ ,  $y = 4$ ) and target (e.g.  $x = 3$ ,  $y = 5$ ) nodes of a link and a constant bias value. The CPPN takes these five values as inputs and produces two output values. The first output value determines the weight of the link between the associated input (source) and hidden layer (target) nodes, and the second output value determines the weight of the link between the associated hidden (source) and output (target) layer nodes. All pairwise combinations of source and target nodes are iteratively passed as inputs to a CPPN to determine the weight of each ANN link.

HyperNEAT can exploit the geometry of a problem because the link values between nodes in the ANN phenotype are a function of the geometric positions of those nodes (Stanley et al., 2009; Clune et al., 2009c, 2011). For quadruped locomotion, this property has been shown to help HyperNEAT produce gaits with front-back, left-right, and four-way symmetries (Clune et al., 2009a, 2011).

The evolution of the population of CPPNs occurs according to the principles of the NeuroEvolution of Augmenting Topologies (NEAT) algorithm (Stanley and Miikkulainen, 2002), which was originally designed to evolve ANNs. NEAT can be fruitfully applied to CPPNs because of their structural similarity to ANNs. For example, mutations can add a node, and thus a function, to a CPPN graph, or change its link weights. The NEAT algorithm is unique in three main ways (Stanley and Miikkulainen, 2002). Initially, it starts with small genomes that encode simple networks

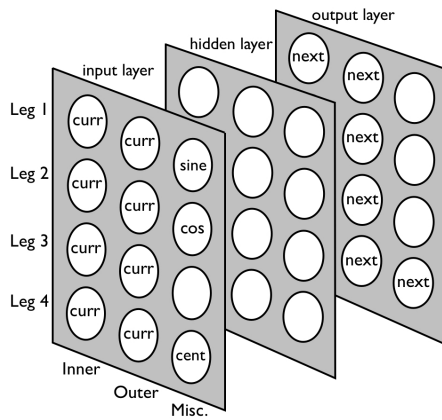


Figure 6: ANN configuration for HyperNEAT runs. The first two columns of each row of the input layer receive information about a single leg (the angles requested in the previous time step for its two joints). The final column provides the previously requested angle of the center joint and, to enable periodic movements, a sine and cosine wave. Evolution determines the function of the hidden-layer nodes. The nodes in the output layer specify new joint angles for each respective joint. The unlabeled nodes in the input and output layers are ignored. Figure adapted from Clune et al. (2011).

and slowly complexifies them via mutations that add nodes and links to the network, enabling the algorithm to evolve the topology of an ANN in addition to its weights. Secondly, NEAT has a fitness-sharing mechanism that preserves diversity in the system and gives time for new innovations to be tuned by evolution before competing them against more adapted rivals. Finally, NEAT tracks historical information to perform intelligent crossover while avoiding the need for expensive topological analysis. A full explanation of NEAT can be found in (Stanley and Miikkulainen, 2002).

The ANN configuration follows previous studies that evolved quadruped gaits with HyperNEAT in simulation (Clune et al., 2011, 2009a), but was adapted to accommodate the physical robot in this paper. Specifically, the ANN has a fixed topology (i.e. the number of nodes does not evolve) that consists of three  $3 \times 4$  Cartesian grids of nodes forming input, hidden, and output layers (Figure 6). Adjacent layers were allowed to be completely connected, meaning that there could be  $(3 \times 4)^2 = 288$  links in each ANN (although evolution can set weights to 0, functionally eliminating the connection). The inputs to the substrate were the angles *requested* in the previous time step for each of the 9 joints of the robot (recall that gaits are open-loop, so actual joint angles are unknown) and a sine and cosine wave (to facilitate the production of periodic behaviors). The sine and cosine waves had a period of about half a second.

The outputs of the substrate at each time step were nine numbers in the range  $[-1, 1]$ , which were scaled according

to the allowable ranges for each of the nine motors and then commanded the positions for each motor. Occasionally HyperNEAT would produce networks that exhibited rapid oscillatory behaviors, switching from extreme negative to extreme positive numbers each time step. This resulted in motor commands to alternate extremes every 25ms (given the command rate of 40Hz), which tended to damage and overheat the motors. To ameliorate this problem, we requested four times as many commanded positions from HyperNEAT ANN's and averaged over four commands at a time to obtain the actual gait  $g(t)$ . This solution worked well and did not restrict the expressiveness of HyperNEAT.

As with the parameterized methods, three runs of HyperNEAT were performed. Runs lasted 20 generations with a population size of 9 organisms in 3 species, allowing a bare minimum of diversity within and between NEAT species. These numbers were necessarily small given how much time it took to conduct evolution directly on a real robot. The remaining parameters were identical to Clune et al. (2011).

## Results and Discussion

### Learning Methods for Parameterized Gaits

The results for the parameterized gaits are shown in Figure 7 and Table 2. A total of 1217 hardware fitness evaluations were performed during the learning of parameterized gaits, with the following distribution by learning method: 200 random, 234 uniform, 284 Gaussian, 174 gradient, 172 simplex, 153 linear regression. The number of runs varies because each run plateaued at its own pace. The best overall gait for the parameterized methods was found by linear regression, which also had the highest average performance. The Nelder-Mead simplex also performed quite well on average. The other local search methods did not outperform random search; however, all methods did manage to explore enough of the parameter space to significantly improve on the previous hand-coded gait in at least one of the three runs. No single strategy consistently beat the others: for the first trial Linear Regression produced the fastest gait at 27.58 body lengths/minute, for the second a random gait actually won with 17.26, and for the third trial the Nelder-Mead simplex method attained the fastest gait with 14.83.

One reason the randomly-generated SineModel5 gaits were so effective may have been due to the SineModel5's bias toward regular, symmetric gaits. This may have allowed the random strategy — focusing on exploration — to be competitive with the more directed strategies that exploit information from past evaluations.

### HyperNEAT Gaits

The results for the gaits evolved by HyperNEAT are shown in Figure 8 and Table 2. A total of 540 evaluations were performed for HyperNEAT (180 in each of three runs). Overall the HyperNEAT gaits were the fastest by far, beating all the parameterized models when comparing either average

	Average	Std. Dev.
Previous hand-coded gait	5.16	—
Random search	9.40	6.83
Uniform Random Hill Climbing	7.83	4.56
Gaussian Random Hill Climbing	10.03	6.00
Policy Gradient Descent	6.32	7.39
Nelder-Mead simplex	12.32	3.35
Linear Regression	14.01	12.88
Evolved Neural Network (HyperNEAT)	29.26	6.37

Table 2: The average and standard deviation of the best gaits found for each algorithm during each of three runs, in body lengths/minute.

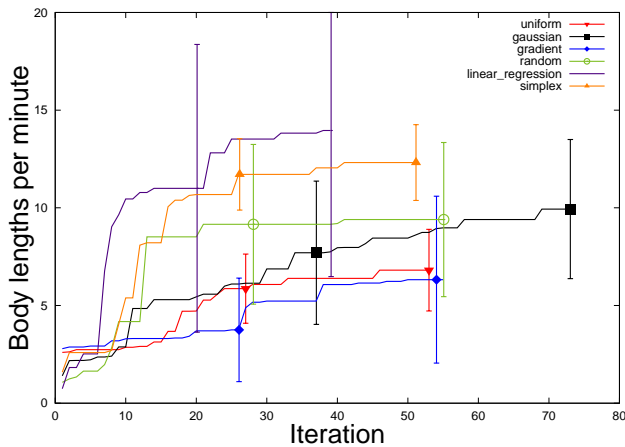


Figure 7: Average results ( $\pm$  SE) for the parameterized learning methods, computed over three separately initialized runs. Linear regression found the fastest overall gait and had the highest average, followed by Nelder-Mead simplex. Other methods did not outperform a random strategy.

or best gaits. We believe that this is because HyperNEAT was allowed to explore a much richer space of motions, but did so while still utilizing symmetries when advantageous. The single best gait found during this study had a speed of 45.72 body lengths/minute, 66% better than the best non-HyperNEAT gait and 8.9 times faster than the hand-coded gait. Figure 9 shows a typical HyperNEAT gait that had high fitness. The pattern of motion is both complex (containing multiple frequencies and repeating patterns across time) and regular, in that patterns of multiple motors are coordinated.

The evaluation of the gaits produced by HyperNEAT was more noisy than for the parameterized gaits, which made learning difficult. For example, we tested an example HyperNEAT generation-champion gait 11 times and found that its mean performance was 26 body lengths/minute ( $\pm$  13 SD), but it had a max of 38 and a min of 3. Many effective HyperNEAT gaits were not preserved across generations because

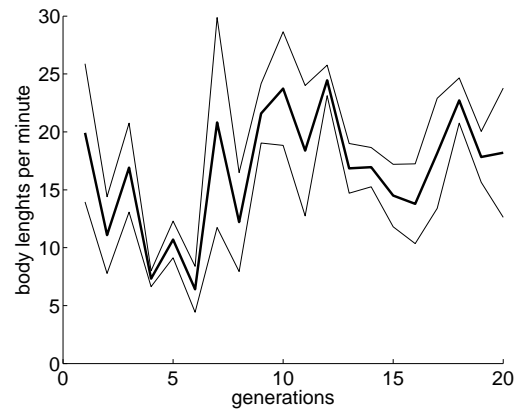


Figure 8: Average fitness ( $\pm$  SE) of the highest performing individual in the population for each generation of HyperNEAT runs. The fitness of many high-performing HyperNEAT gaits were halved if the gait overly stressed the motors (see text), meaning that HyperNEAT’s true performance without this penalty would be even higher.

a single poor-performing trial could prevent their selection. The HyperNEAT learning curve would be smoother if the noise in the evaluations could be reduced or more than one evaluation per individual could be afforded.

## Conclusion and Future Work

We have presented an array of approaches for optimizing a quadrupedal gaits for speed. We implemented and tested six learning strategies for parameterized gaits and compared them to gaits produced by neural networks evolved with the HyperNEAT generative encoding.

All methods resulted in an improvement over the robot’s previous hand-coded gait. Building a model of gait performance with linear regression to predict promising directions for further exploration worked well, producing a gait of 27.5 body lengths/minute. The Nelder-Mead simplex method performed nearly as well, likely due to its robustness to noise. The other parameterized methods did not outperform random search. One reason the randomly-generated SineModel5 gaits performed so well could be because the gait representation was biased towards effective, regular gaits, making the highly exploratory random strategy more effective than more exploitative learning algorithms.

HyperNEAT produced higher-performing gaits than all of the parameterized methods. Its best-performing gait traveled 45.7 body lengths per minute, which is nearly 9 times the speed of the hand-coded gait. This could be because HyperNEAT tends to generate coordinated gaits (Clune et al., 2011, 2009a), allowing it to take advantage of the symmetries of the problem. HyperNEAT can also explore a much larger space of possibilities than the more restrictive 5-dimensional parameterized space. HyperNEAT gaits



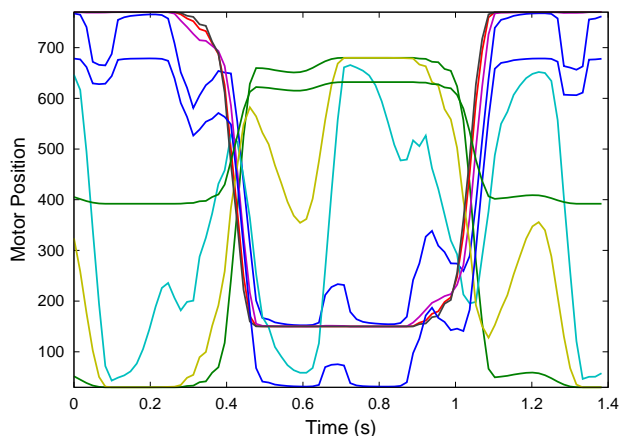


Figure 9: Example of one high-performance gait produced by HyperNEAT showing commands for each of nine motors. Note the complexity of the motion pattern. Such patterns were not possible with the parameterized SineModel5, nor would they likely result from a human designing a different low-dimensional parameterized motion model.

tended to produce more complex sequences of motor commands, with different frequencies and degrees of coordination, whereas the parameterized gaits were restricted to scaling single-frequency sine waves and could only produce certain types of motor regularities.

Because all 1217 trials were done in hardware, it was difficult to gather enough data to properly rank the methods statistically. One direction for future work could be to obtain many more trials. However, a more effective extension might be to combine frequent trials in simulation with infrequent trials in hardware (Bongard et al., 2006). The simulation would produce the necessary volume of trials to allow the learning methods to be effective, and the hardware trials would serve to continuously ground and refine the simulator. One could also guide evolution to the most fertile territory by penalizing gaits that produced large discrepancies between simulation and reality (Koos et al., 2010). Another extension would be to allow gaits that sensed the position of the robot and other variables to enable the robot to adjust to its physical state, instead of providing an open-loop sequence of motor commands. All of these approaches would likely improve the quality of automatically generated gaits for legged robots, which will hasten the day that humanity can benefit from their vast potential.

### Acknowledgments

NSF Postdoctoral Research Fellowship in Biology to Jeff Clune (award number DBI-1003220).

### References

Bongard, J., Zykov, V., and Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science*, 314(5802):1118–

1121.

Chernova, S. and Veloso, M. (2005). An evolutionary approach to gait learning for four-legged robots. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2562–2567. IEEE.

Clune, J., Beckmann, B., Ofria, C., and Pennock, R. (2009a). Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 2764–2771.

Clune, J., Beckmann, B., Pennock, R., and Ofria, C. (2009b). HybridID: A Hybridization of Indirect and Direct Encodings for Evolutionary Computation. In *Proceedings of the European Conference on Artificial Life*, pages 134–141.

Clune, J., Ofria, C., and Pennock, R. (2009c). The sensitivity of HyperNEAT to different geometric representations of a problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 675–682. ACM.

Clune, J., Stanley, K., Pennock, R., and Ofria, C. (2011). On the performance of indirect encoding across the continuum of regularity. *IEEE Transactions on Evolutionary Computation*. To appear.

Hornby, G., Takamura, S., Yamamoto, T., and Fujita, M. (2005). Autonomous evolution of dynamic gaits with two quadruped robots. *IEEE Transactions on Robotics*, 21(3):402–410.

Kohl, N. and Stone, P. (2004). Policy gradient reinforcement learning for fast quadrupedal locomotion. *IEEE International Conference on Robotics and Automation*, 3:2619–2624.

Koos, S., Mouret, J., and Doncieux, S. (2010). Crossing the reality gap in evolutionary robotics by promoting transferable controllers. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 119–126. ACM.

Singer, S. and Nelder, J. (2009). Nelder-mead algorithm. [http://www.scholarpedia.org/article/Nelder-Mead\\_algorithm](http://www.scholarpedia.org/article/Nelder-Mead_algorithm).

Stanley, K. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162.

Stanley, K., D’Ambrosio, D., and Gauci, J. (2009). A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212.

Stanley, K. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.

Téllez, R., Angulo, C., and Pardo, D. (2006). Evolving the walking behaviour of a 12 dof quadruped using a distributed neural architecture. *Biologically Inspired Approaches to Advanced Information Technology*, pages 5–19.

Valsalam, V. and Miikkulainen, R. (2008). Modular neuroevolution for multilegged locomotion. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 265–272. ACM.

Zykov, V., Bongard, J., and Lipson, H. (2004). Evolving dynamic gaits on a physical robot. *Proceedings of Genetic and Evolutionary Computation Conference, Late Breaking Paper, GECCO*, 4.

# Landmark Vector Model with Quantized Distance for Homing Navigation

Seung-Eun Yu<sup>1</sup> and DaeEun Kim<sup>1</sup>

<sup>1</sup>Biological Cybernetics Lab, Yonsei University, Seoul, South Korea  
{s.eun, daeeun}@yonsei.ac.kr

## Abstract

Inspired by astonishing navigation ability of insects and other animals, many studies observed their behaviors, and considered biomimetic application to robotic systems by investigating mechanisms based on various senses. In this paper, we suggest a new landmark vector model for homing navigation with quantized distance information. The method is highly successful for homing navigation in both perspectives of angular error and success rate. This work has been published in Yu and Kim (2011).

## Introduction

Animals have developed navigation skills based on various senses. Desert ants and honeybees are known to use visual information (Collett, 1996), turtles migrating long distance rely on magnetic compass (Luschi et al., 1996), while other studies have shown that birds use olfactory cues to navigate (Papi, 1990). Many studies have focused on designing bio-inspired navigation algorithms for robotic systems inspired by the excellent performance animals demonstrate. Among them, vision-based homing navigation has been studied through a number of bio-inspired algorithms. One of the simplest method suggested, inspired by desert ants and honeybees, is the ‘snapshot model’ (Cartwright and Collett, 1983).

In the snapshot model, currently obtained visual information is compared to that in the snapshot image taken at home location. Several different methods were suggested to process the snapshot images for homing navigation. One of the methods based on the concept is the average landmark vector (ALV) model by Lambrinos et al. (2000).

The ALV model is one parameter method where the average landmark vector is obtained by averaging every unit-length landmark vectors perceived in the snapshot. Comparing the average landmark vectors obtained from snapshots taken at the current location and at home location, it is sufficient to guide the agent for the homing direction. The ALV model is based on a simple representation of the environment with powerful performance results in homing navigation, but the model necessarily requires a reference compass information.

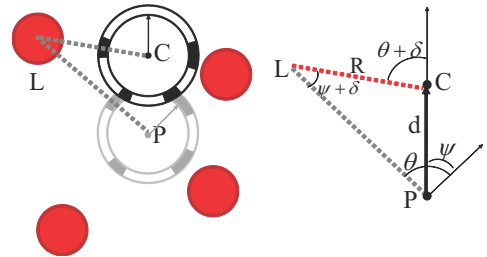


Figure 1: Image shift of landmarks when the agent moves from the position  $P$  to  $C$  with moving distance  $d$ . The head orientation angle changes by  $\psi$ , and the viewing angle of a landmark is  $\theta$  and  $\theta + \delta$  in two positions, respectively (modified from Yu and Kim (2010)).

In this paper, we propose a new landmark-based navigation algorithm without any reference compass. The method we suggest is the distance-estimated landmark vector model (DELV) using quantized distance estimation along with the rotational landmark arrangement matching. This work has been published in Yu and Kim (2011).

## Methods

While the ALV model (Lambrinos et al., 2000) considers landmark vectors in unit length, and perceives only angular directions of landmarks ignoring their distances, the DELV model includes distance information as well as the angular position of landmarks in the landmark vector. Both methods share similar concept in perceiving landmark information as a vector form, but the matching process between information in two snapshots have a different point of view in exploiting the landmark vectors.

### Distance estimation and quantization

The DELV method includes distance information of landmarks in the landmark vector representation, which can be obtained by inducing the image motion. Using an omnidirectional camera, the mobile robot is able to monitor 360° view of its surroundings, and the angular position of landmarks observed in the view is shifted as the robot moves

one step forward. Fig. 1 describes the geometric relationship between angular shift and the distance, and Eq. 1 shows the distance estimation based on the relationship.

$$R = \frac{d \sin(\theta - \psi)}{\sin(\delta + \psi)} \quad (1)$$

As in Eq. 1, the estimation of landmark distance is affected by variables such as  $\theta$ ,  $\delta$  and  $d$ , and their accuracies can affect the estimation results. The angular position of landmarks  $\theta$  and  $\theta + \delta$  is sensitive to noise in the captured image, while the moving distance  $d$  is influenced by odometry error. In addition, it may be plausible to argue that insects or other animals perceive the distances to landmarks in a relative manner rather than in the absolute values. Therefore, we apply quantization on the estimated distance to landmark vectors. Through arrangement matching of landmark vectors for heading direction estimation and homing direction computation, it is shown that the landmark vector model with quantized distance is effective, which will be described in more details.

### Rotational matching of landmark vectors

The landmark vector set perceived at home location is stored as a reference map in which the landmark vectors at an arbitrary location will be projected to obtain a homing direction. By reversely projecting the landmark vector obtained at the current location to the reference map, the end point of each landmark vector would represent the vector from the home location to the current location. With  $N$  landmarks available in the environment, the estimation on the current position  $p^k(x)$  is defined as the average of the landmark vectors projected on the reference map. Assuming the appropriate arrangement  $k$ , the equation is given as:

$$p^k(x) = \frac{1}{N} \sum_{i=1}^N [V_i^R(x_o, \alpha_r) - V_i^k(x, \alpha)] \quad (2)$$

where  $V_i^R(x_o, \alpha_r)$  is the landmark vector for the  $i$ -th landmark in the reference map, and  $V_i^k(x, \alpha)$  is the  $i$ -th landmark vector with the matching order  $k$  at the current location  $x$ .

Without a reference compass, the DELV method solves the correspondence problem between landmarks in a pair of snapshots with the rotational arrangement matching of landmark vectors. The variance of end points of the projected landmark vectors is used as the criterion for finding the best matching order  $z$  and heading direction  $\alpha_z$  as:

$$\arg \min_{k, \alpha} \left[ \sum_{i=1}^N \left( [V_i^R(x_o, \alpha_r) - V_i^k(x, \alpha) - p^k(x)]^T [V_i^R(x_o, \alpha_r) - V_i^k(x, \alpha) - p^k(x)] \right) \right] \quad (3)$$

### Results and discussion

Vector map results in Fig. 2 (a) to (c) indicate the decided homing direction with the suggested method for three different quantization levels. Angular error curves in Fig. 2 (d) efficiently compares

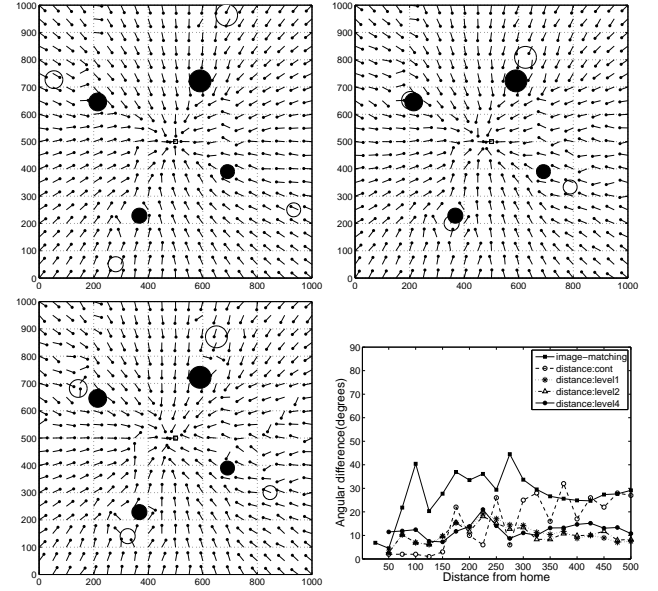


Figure 2: Vector map results with quantization in (a) level 1, (b) level 3, and (c) level 5, respectively and (d) error curves

the results and show low angular errors. As a result, the DELV model with quantized distance shows homing ability with simple representation of environments and low complexity in computation even without any reference compass information. The quantization of distances to landmarks may allow some errors in heading direction search and current location estimation, but experimental results showed that the method leads to the homing direction decision.

### Acknowledgements

This work was supported by the Mid-career Researcher Program through an NRF grant funded by the MEST in South Korea (No. 2011-0000210).

### References

- Cartwright, B. and Collett, T. (1983). Landmark learning in bees. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 151(4):521–543.
- Collett, T. (1996). Insect navigation en route to the goal: multiple strategies for the use of landmarks. *Journal of Experimental Biology*, 199(1):227–235.
- Lambrinos, D., Moller, R., Labhart, T., Pfeifer, R., and Wehner, R. (2000). A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, 30(1-2):39–64.
- Luschi, P., Papi, F., Liew, H., Chan, E., and Bonadonna, F. (1996). Long-distance migration and homing after displacement in the green turtle (*Chelonia mydas*): a satellite tracking study. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 178(4):447–452.
- Papi, F. (1990). Olfactory navigation in birds. *Cellular and Molecular Life Sciences*, 46(4):352–363.
- Yu, S. and Kim, D. (2011). Landmark vectors with quantized distance information for homing navigation. *Adaptive Behavior*, 19(2):121–141.
- Yu, S.-E. and Kim, D. (2010). Discretized landmark distance estimation method for mobile robot navigation. *Korea Robotics Society Annual Conference 2010*, pages 136–137.



# On the interaction of adaptive timescales on networks

Elisabeth zu Erbach-Schoenberg<sup>1,2</sup>, Connor McCabe<sup>3</sup> and Seth Bullock<sup>1,3</sup>,

<sup>1</sup>Institute for Complex Systems Simulation, <sup>2</sup>School of Management, and

<sup>3</sup>School of Electronics and Computer Science, University of Southampton, Southampton, UK  
e.zuerbach-schoenberg@soton.ac.uk

## Abstract

The dynamics of real-world systems often involve multiple processes that influence system state. The timescales that these processes operate on may be separated by orders of magnitude or may coincide closely. Where timescales are not separable, the way that they relate to each other will be important for understanding system dynamics. In this paper, we present a short overview of how modellers have dealt with multiple timescales and introduce a definition to formalise conditions under which timescales are separable.

We investigate timescale separation in a simple model, consisting of a network of nodes on which two processes act. The first process updates the values taken by the network's nodes, tending to move a node's value towards that of its neighbours. The second process influences the topology of the network, by rewiring edges such that they tend to more often lie between similar individuals. We show that the behaviour of the system when timescales are separated is very different from the case where they are mixed. When the timescales of the two processes are mixed, the ratio of the rates of the two processes determines the system's equilibrium state. We go on to explore the impact of heterogeneity in the system's timescales, i.e., where some nodes may update their value and/or neighbourhood faster than others, demonstrating that it can have a significant impact on the equilibrium behaviour of the model.

## Introduction

Real-world adaptive systems typically involve many interacting parts and processes operating at multiple timescales. However, models of these systems often proceed by identifying a single substantive timescale. Faster processes are often idealised as essentially instantaneous, while slower processes are often treated as a constant background influence that parametrises the model's dynamics.

For instance, Kauffman's (1993) NK landscape model of adaptation on rugged fitness landscapes has a single substantive timescale. At each step the genotype of a genetically fixated population is updated to one of the fitter adjacent genotypes. In reality, a newly discovered fitter mutant takes time to reach fixation. This process is idealised as instantaneous. During a single run of Kauffman's model, the parameters  $N$

and  $K$ , which determine the length of a genome and the degree of epistasis within it, are held fixed. They parametrise the system's dynamics. Of course in reality both  $N$  and  $K$  vary as a consequence of evolutionary change. The security of Kauffman's idealisations hinges on whether these processes are separable: the faster processes are much faster, and the slower processes much slower, than the timescale of the process that he focuses on. There are various interpretations of this kind of concept and in the scope of this paper we define *separation of timescales* as follows. The timescales of two processes are separated if one process leads the system into equilibrium before the other process influences the system. This means when the second process sets in, the system has already reached equilibrium.

Where processes take place over similar timescales and affect each other, i.e., they are coupled, dealing with these interacting timescales becomes an important issue. For real-world systems there are further considerations that may be significant. To what extent is there component-wise *heterogeneity* in the rates at which different components operate? While, on average, a genome's alleles might mutate with probability  $p$ , it may be the case that some alleles are more vulnerable to mutation than others. While, on average, the children in a schoolyard might update their social ties at rate  $r$ , some might update these ties more often than others. Moreover, a system's timescales might vary with time. The traffic on a network might be diurnal, with higher rates during the day. The rate of plasticity in a neural component might decay with the age of the component.

Here we are interested in exploring these issues in the context of adaptive processes modelled on networks. In the co-evolutionary networks literature (Gross and Blasius, 2008), two processes are typically modelled: one governing the tendency for nodes to change their *state*, and one governing *topological change* in the network. These two processes may occur on separable timescales where interactions between the two processes can perhaps be neglected. If the timescales for the two processes are not separable, their interplay will affect the behaviour exhibited by the network.

Here we explore a very simple coevolutionary network in

which both state and topology evolve over time. We first vary the rates of change for both processes and demonstrate that their ratio impacts on the equilibrium state of the network. We proceed to explore the impact of heterogeneity in timescale, demonstrating that it can impact on both the distribution of node states and the topology at equilibrium. Before introducing the simple model and its results, we review some literature demonstrating the issue of timescale in modelling adaptive systems. We conclude with discussion of the results presented here and ideas for future work.

## Dealing with timescale

### Synchrony vs. asynchrony

Several studies have revealed that models of adaptive systems can be sensitive to the updating scheme chosen. Using a synchronous model of the iterated prisoner's dilemma, Nowak and May (1992) found complicated spatial patterns within which co-operation persisted. Using an asynchronous update scheme for the same model, Huberman and Glance (1993) found that spatial patterns disappeared with defection the only strategy adopted. While Kauffman (1993) has shown that synchronous Random Boolean Networks can exhibit many stable cyclic behaviours, glossed as analogous to the multiple cell types that may result from the same genome, Harvey and Bossomaier (1997) showed that the same Random Boolean Networks with asynchronous update would tend to evolve to a fixed point.

### Multiple timescales

Artificial life has typically considered multiple adaptive timescales in the context of interactions between learning and evolution (Ackley and Littman, 1992; Belew and Mitchell, 1996), such as the Baldwin effect (Hinton and Nowlan, 1987). Further examples where the separation of timescales is critical to adaptive dynamics include the interaction between processes of neurotransmission and (much slower) neuromodulation (Buckley et al., 2004, 2005; Buckley, 2008; Husbands et al., 2010), and the interaction between the evolution of individual behaviours and ecological relationships (e.g., Powers et al., in press; Watson et al., in press; Van Der Laan and Hogeweg, 1995).

### Timescales on networks

Most research involving dynamic networks has focused on addressing either the dynamics 'on' a network, or the dynamics 'of' the network (Gross and Blasius, 2008). The dynamics 'on' a network describe the state transitions of the network's nodes, while the dynamics 'of' a network describe topological changes. Research on so-called coevolutionary networks recognises that these processes are inherently reflexive, with network state influencing topological change (as when edges are formed between similar nodes), and topology constraining state change (as when neighbours exchange information) (Blasius and Gross, 2009; Gross and

Blasius, 2008; Gross and Sayama, 2009). Coevolutionary networks have been the subject of recent study in the context of the epidemic spread of diseases (Newman, 2002; Zhong et al., 2010; Funk and Jansen, 2010; Van Segbroeck et al., 2010), cascading network behaviour (Watts, 2002), opinion dynamics (Kozma and Barrat, 2008; Demirel et al., 2011), diffusion of innovations / information (Onnela and Reed-Tsochas, 2009; Ke and Yi, 2008), evolution of social groups (Palla et al., 2007), the growth of social networks (Sun and Wang, 2008), co-operation (Pacheco et al., 2006; Van Segbroeck et al., 2009), community formation (Bryden et al., 2010), synchronisation (Zhu et al., 2010) and global adaptation (Watson et al., in press). The dynamical interplay of state update and rewiring processes are typically central to the evolution of these systems.

### Heterogeneous timescales

Typically, models make a simplifying assumption that all components update their state at a shared characteristic rate, while structural relationships change at some other arbitrary rate. However, some models have explored systems with heterogeneous rates. Van Segbroeck et al. (2009), for instance, found that increased diversity in their model accelerates the rate of evolution to an equilibrium state where co-operation is a robust and dominant strategy. Pacheco et al. (2006) employed variable re-wiring rates in a social agent model. Their results suggest that introducing heterogeneity has an effect on the system as a whole which can change the frequency of co-operation observed at equilibrium.

### A simple model

To study the influence of timescale separation we introduce an abstract model based on models of opinion dynamics that include adaptive change in network topology as well as the spread of opinions over the network (e.g., Kozma and Barrat, 2008). Here, nodes have an internal value and tend to update this value in the direction of their neighbours' values. The second process changes the network topology by rewiring edges between nodes such that nodes disconnect from dissimilar neighbours and connect to nodes with more similar values.

To illustrate what kind of processes this model could be related to we could assume that each node's value represents the opinion of a different person and that edges represent social interactions between people. In this setup, we can imagine that either rewiring or state update might be the faster process. If we assume a node's value represents something such as the religion a person believes in or a political affiliation, we can assume that this value changes very slowly. We can further assume that therefore a person would more readily change to associate with individuals sharing a similar opinion than change their own opinion to match that of their neighbours. In this case, the rewiring process would be faster than the value update process. At the other end of

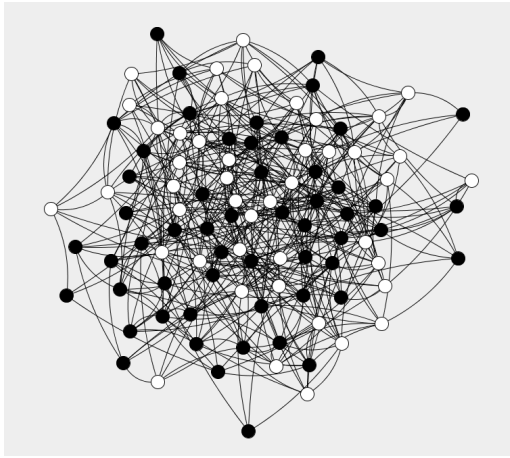


Figure 1: A typical network after initialisation.

the spectrum, we could assume a node's value represented a person's preference for meeting friends at one restaurant rather than another. In this case the individuals would be likely to change their opinion based on the opinions of their friends, rather than changing their friends on the basis of their restaurant preference. In between these two extremes we can think of intermediate cases where individuals have a preference for socialising with individuals that share a similar opinion, but also change their own opinion towards that of their neighbours.

### The model

The model consists of a network of  $N$  interconnected nodes (here  $N = 100$ ). Each node has a single value in the interval  $[0.0, 1.0]$ . Even though a node's value can be any value between 0 and 1, each starts with the value 0.0 or 1.0, with equal probability. Nodes are connected by undirected, unweighted edges, meaning an edge is either present or absent and if node  $a$  cuts a tie to  $b$ ,  $b$  also loses its connection to  $a$ . Self-connections are not allowed. To initialise the network between the nodes, we specify an average degree  $d$  and generate a random network by making an edge between each possible pair of unique nodes with probability  $\frac{d}{N-1}$ . In the examples presented here, we use an average degree of  $d = 10$ . A visualisation of a typical network after initialisation is given in Figure 1.

**Value Update:** When a node  $i$  updates its value, it chooses a random individual  $n$  from the set of its neighbours. It then discovers the value of its neighbour  $v(n)$  and calculates the difference  $v(n) - v(i)$  between the neighbour's value and its own. The node then updates its state towards the state of its neighbour, proportional to the difference in values:  $v(i)_{t+1} = v(i) + m(v(n) - v(i))$ . The factor  $m$  determines the maximal change that can occur in one step. Here we choose  $m = 0.01$ , to ensure that it takes several updates for two nodes to reach the same value. If the updating node and

its chosen neighbour have the same state, i.e.,  $v(i) = v(n)$ , the update results in no change to  $v(i)$ .

**Rewiring:** When node  $i$  rewires, it compares its own value to the values of its neighbours, identifying the neighbour with which it is most dissimilar,  $n$ . The node  $i$  then generates a list of all neighbours of all of its neighbours, comprising all nodes that are two edges away. Members of this list that are already neighbours of  $i$  are discarded. If the list is non-empty,  $i$  drops the connection to  $n$  and rewires this edge to a randomly chosen member of the list of neighbours' neighbours. This implies that, if an individual is already connected to all neighbours of its direct neighbours, an attempt to rewire will result in no topological change.

**Timescales:** In each step of the algorithm, a list is generated containing all nodes that are ready to update their state in the current time step. These nodes are then updated in a random order, one at a time. After this, the same procedure is repeated for all nodes ready to rewire. Whether a node is ready to update or rewire depends on the timescales of the two processes. The relation of the timescales is incorporated in the model as follows. Each node is assigned two values,  $V_i$  and  $R_i$ , specifying the number of time steps in the interval between two consecutive value updates for  $i$  and two consecutive rewiring events for  $i$ , respectively. In the case of homogeneous timescales, all nodes have identical values for  $V$  and identical values for  $R$ , i.e.,  $\forall i V_i = V$  and  $\forall i R_i = R$ . In the case of heterogeneous timescales this constraint does not hold and values for the two rates may differ from node to node. The algorithm stops when neither the state update nor rewiring process effects any change in the network. We will consider this stopping criterion in more detail next.

**Equilibration:** Both the value update and the rewiring process can only change the system's state if there is a local difference between two nodes. A local difference is present if two nodes that are connected by an edge have non-identical values. This difference can be reduced by updating the value of one or both nodes or by deleting the edge between the two nodes and rewiring it to a node with a more similar value. Once there are no local differences in the system anymore, neither the value update process nor the rewiring process change the system's state when invoked. Therefore both processes need a value difference between connected nodes to operate. Thus, we can see the difference in values between connected nodes as some kind of energy available to the two processes to use for changing the system's state. Both processes can only operate if there is energy left in the system and both processes reduce the energy, at least locally. One way of formally defining this energy is as the sum of absolute value differences between all pairs of connected nodes,  $e = \sum_{i,j \text{ connected}} |v(i) - v(j)|$ .

The energy specified in this way reduces over time and

once it has reached zero, the system's state cannot change any more. Therefore, we can use reaching zero energy as a formal stopping criterion and terminate the algorithm when the energy has reached zero. Note that, from the initial conditions considered here, each process is capable of reducing energy to zero in the absence of the other.

In the case of homogeneous timescales, whether they are separated depends on the ratio of the values  $V$  and  $R$ . For  $R \gg V$  the timescales of the two processes are separated, with only the value update process influencing the dynamics. We also have separation of timescales in the opposite case,  $V \gg R$ , where the rewiring process dominates the dynamics. Let us now specify further when exactly the timescales are separated to find values for the parameters  $V$  and  $R$  for which we can be certain the timescales are separate. Based on the definition presented in the introduction, the timescales of the two processes are separate if one process acts after the other process has reached equilibrium. Based on the equilibrium definition as a zero energy state, we define the equilibrium points  $t_R$  and  $t_V$  as the number of steps the rewiring or value update process takes in isolation to reduce the energy of the system to zero and therefore reach equilibrium. We measure these two points for a particular set of initial conditions by running the algorithm with only one of the two processes operating. Measuring the time the system takes to reach zero energy when only one process acts on it is the equilibrium time for that process,  $t_V$  or  $t_R$ . If the second process acts only after the system has reached equilibrium, it is unable to change the system state as there is no energy for it to exploit (i.e., no value difference between connected nodes). This means that the timescales of the two processes are separated in two cases. The first case is when  $V > t_R$ , meaning that the value update only happens after the rewiring process has brought the system to equilibrium. In the second case, for  $R > t_V$ , the rewiring process happens after the value update process has already reduced the system's energy to zero. In any other case the timescale are mixed to some degree.

## Results

We now observe the system behaviour for varying ratios  $\frac{R}{V}$ , first for homogeneous timescales and then for varying degrees of heterogeneity.

### Homogeneous timescales

If only the rewiring process is active and its rate is the same for all nodes, the system reaches equilibrium after  $t_R \approx 10.0$  steps. If only the state update process is active, it takes longer for the system to reach equilibrium,  $t_V \approx 6500$ . Having measured these values, we can assign values to the parameters  $V$  and  $R$  for which the timescales are separated and one of the two processes dominates the dynamics.

Setting  $V = 100$  and  $R = 1$ <sup>1</sup> the timescales are separated

<sup>1</sup>Note that the equilibrium times measured above assume that

as  $V > t_R = 10$ , with only the rewiring process influencing the dynamics as it reaches equilibrium before the state change process has time to affect the network.

The equilibrium state of the system under these parameters is shown in Figure 2a<sup>2</sup>. Since the network is initially populated by equal numbers of nodes with value 0.0 and value 1.0, the rewiring process removes edges between dissimilar nodes and replaces them with edges linking nodes with identical value, forming two homogeneous components, one containing all the nodes with value 0.0 and the other containing the nodes initialised with value 1.0. At the other extreme,  $V = 1$  and  $R = 10000 > t_v$ , only the state update process shapes the network. Figure 2j depicts the equilibrium state under these conditions. Node values have gradually changed towards the average value of the initial population until all nodes have exactly this value. Since all the nodes have identical values no rewiring can take place and the network topology does not change at all.

Intermediate cases where the timescales are mixed are shown in Figures 2b–2i. Where the rewiring process is fast relative to the state update process, the network breaks up into several components, each eventually consisting of nodes with the same value, but with values differing significantly between the components (see, e.g., Figure 2b). Where the system's dynamics are more influenced by the state update process (see, e.g., Figure 2e) the values adopted by different components tend to be less diverse and closer to the system mean. Eventually, the state update dynamic is fast enough to equilibrate the network before the rewiring process can cause it to fragment (see, Figures 2h–2j).

These results show that the system reaches the predicted equilibrium when the timescales are separated. For the intermediate cases with mixed timescales however, the ratio between the two timescales determines which equilibrium the system ends up in and the character of this equilibrium, in terms of the node values and the network topology.

Figure 3 depicts how the distribution of node values at equilibrium varies with  $\frac{R}{V}$ . It shows that for very low values of  $\frac{R}{V}$ , the rewiring process dominates the system dynamics and only the initial values (0.0 and 1.0) are present. As  $\frac{R}{V}$  increases, we observe more and more intermediate values, converging to the average value in the system. For high values of  $\frac{R}{V}$ , there is only one value present in the system, corresponding to the mean of the system's initial values.

A similar transition can be observed for the topology of the network. Figure 4 depicts how the distribution of component sizes at equilibrium varies with  $\frac{R}{V}$ . Here we observe that when rewiring dominates, the two network components have nearly the same size, consisting of roughly half of the nodes each (one is larger as a consequence of the ini-

the process considered happens each time step ( $V = 1$  or  $R = 1$ ). We therefore set the frequency of the faster process to 1.

<sup>2</sup>In the examples presented here, the same initial network shown in Figure 1 is used.

tial random allocation of value to the population of nodes). For mixed timescales, components are smaller and isolated nodes (with component size 1) exist. As we move towards

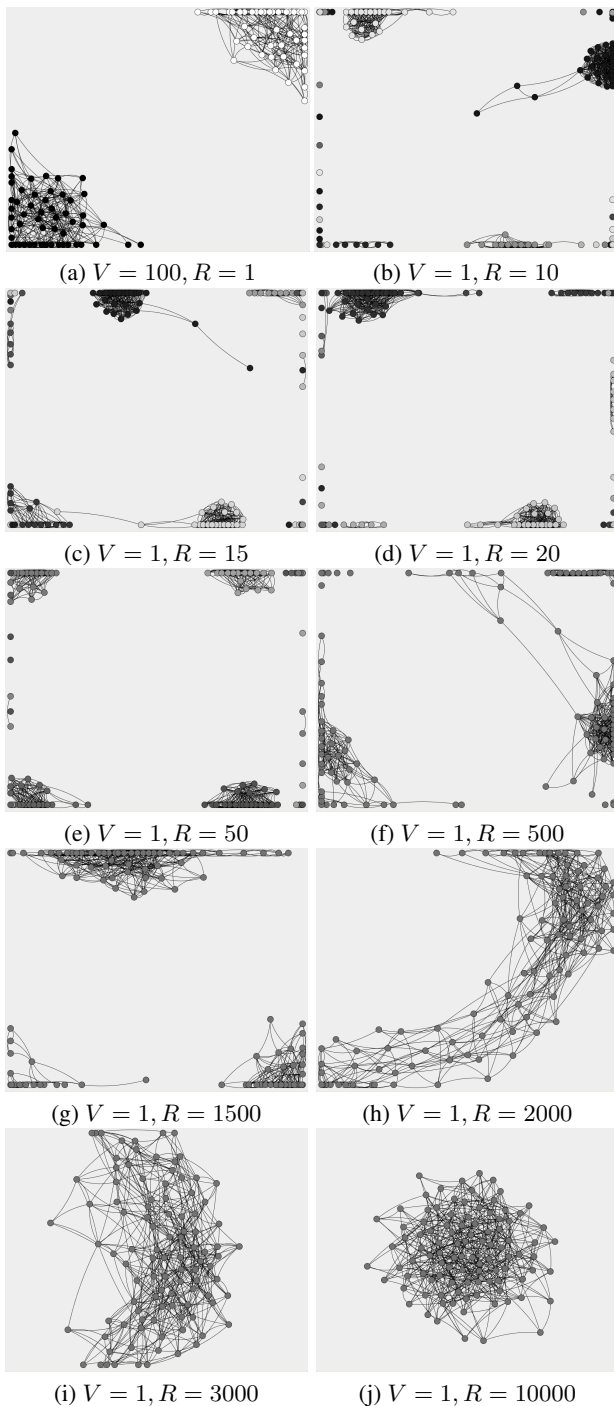


Figure 2: Networks at equilibrium for different values of  $R$  and  $V$ . Node shading indicates the nodes states, with the heaviest shading indicating 0.0 and no shading indicating 1.0.

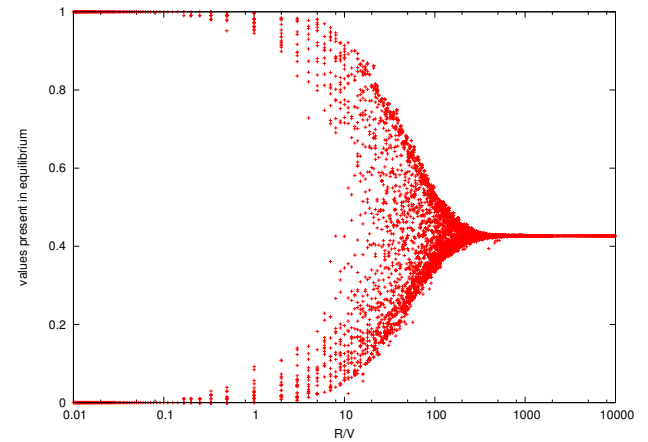


Figure 3: Values present in the equilibrium state for different ratios  $\frac{R}{V}$ . For each value the system is started with the same initial conditions.

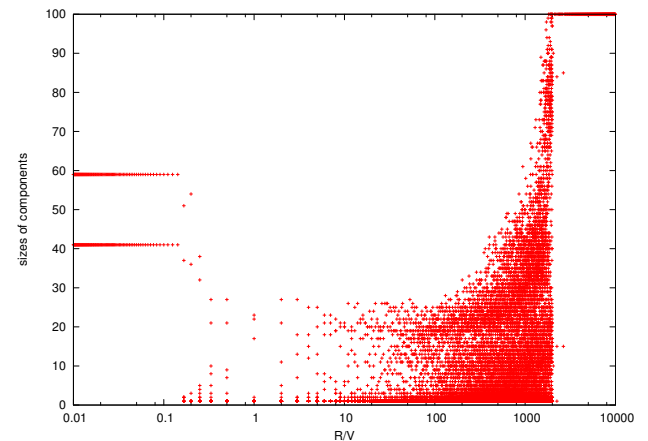


Figure 4: Component sizes present in the equilibrium state for different ratios  $\frac{R}{V}$ . For each value the system is started with the same initial conditions.

the regime where the state update process dominates the systems dynamics, larger components exist at equilibrium. Once state update is the only active process, only one connected component is present at equilibrium.

Comparing these two graphs, we observe that the apparent thresholds in system behaviour exhibited by node values and network topology are different. From the perspective of node values, we can see three regimes separated by two threshold values of  $\frac{R}{V}$ . First, a transition occurs around  $\frac{R}{V} = 0.5$ , with a second qualitative change in the equilibrium behaviour at around  $\frac{R}{V} = 250$ . However, when we consider the network's equilibrium topology, the equivalent transitions seem to occur around  $\frac{R}{V} = 0.1$  and  $\frac{R}{V} = 2500$ .

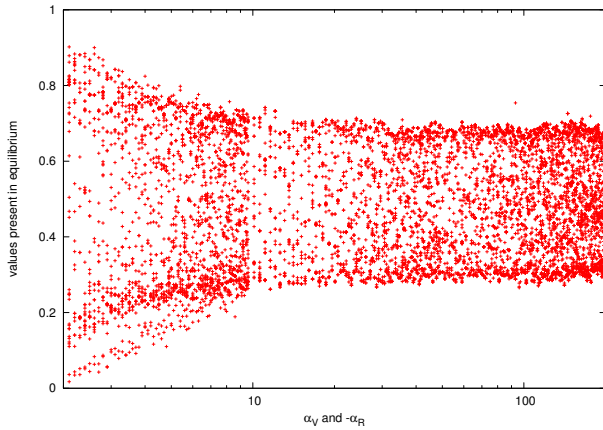


Figure 5: Node values present at equilibrium for different levels of heterogeneity in the value update and rewiring intervals. Only  $\alpha$ , which specifies the heterogeneity of timescales, is varied here (with  $\alpha = \alpha_V = -\alpha_R$ ). The runs are otherwise identical and share the same initial conditions. For each value of  $\alpha$  all node values,  $v(i)$ , present at equilibrium are displayed.

### Heterogeneity in timescales

We now consider the case where some nodes might update their value or their neighbourhood faster than others. We model this by allocating each node,  $i$ , a pair of values,  $V_i$  and  $R_i$ , governing the individual rates of change for value and neighbourhood, respectively. The  $V_i$  and  $R_i$  values are Pareto-distributed, meaning that while most of the values are close to the characteristic population mode,  $V$  or  $R$ , a few are significantly different, due to the long tail of the distribution. Values are generated by transforming a uniform random variable  $U$  by the functions  $\frac{V}{U^{1/\alpha_V}}$  for  $V_i$  and  $\frac{R}{U^{1/\alpha_R}}$  for  $R_i$  (Newman, 2004). The parameters  $\alpha_V$  and  $\alpha_R$  determine the spread of values in each distribution. We set  $\alpha_R$  to a negative value and  $\alpha_V$  to a positive value so that the tails of the distribution point towards each other. Large absolute magnitudes for  $\alpha$  (such as  $\alpha = 100$ ) lead to a relatively small average distance between the resulting values and the modal value,  $V$  or  $R$ , whereas small absolute values for  $\alpha$  (e.g.,  $\alpha = 2.5$ ) produce a larger spread. This introduction of heterogeneity into the model means that each node has its own internal clocks governing when to update its state and when to rewire.

The effect of heterogeneity on the value process is assessed for the case in which  $V = 1$  and  $R = 50$ , as this is an intermediate case where both processes influence the dynamics. Figure 5 shows that for a low degree of heterogeneity in both processes (higher values of  $\alpha$ ) the distribution of values present at equilibrium is not very different from the base case without heterogeneity. For higher levels of heterogeneity, however, the diversity of values increases

significantly. The effects of heterogeneity on the network topology are illustrated in Figures 6 and 7 for  $V = 1$  and  $R = 2000$ , as this ratio of  $\frac{R}{V}$  is the threshold separating single component equilibria from multi-component equilibria. Without heterogeneity, the network forms one component (Figure 7a) with a degree distribution that differs from that of the initial network (compare Figures 6a and 6b). In the presence of heterogeneity however, the network fragments into eleven components (Figure 7b) with a qualitatively different degree distribution (Figure 6c).

### Discussion

The results presented here show that in the cases where the timescales are separated, the system behaves as we would expect: if only the value update process is active, there is no topological change and the values of all nodes converge to the average of the initial network. If only the rewiring process acts on the system state, we only observe changes in topology and the network splits into two components, with nodes being sorted according to their initial value. The number of components in that case depends only on the number of initial values present in the system. For example, if we initialise the system with three (e.g. 0.0, 0.5, 1.0) different values instead of two the network fractures into three clusters. To sum up, when timescales are sufficiently separated, the system behaves in the same way as an equivalent system with the slower process ‘switched off’.

The results also show that if the timescales are not separated, the exact ratio between the rates of the two processes influences the system’s equilibrium state. If the rewiring process dominates the dynamics, the values we find in the system in equilibrium differ significantly. As the value update process gains more influence, the values of the components found in the equilibrium state become more and more similar. We can explain this behaviour by observing the system dynamics over time. Starting from a random initial network, the rewiring process stretches the network into a predominantly white and a predominantly black end. In between, there are nodes of intermediate value. At this stage, if the rewiring is fast, the network fractures at several points. In the case where the value process is the main influence on the system, the values of nodes are more similar at the point when the rewiring sets in, as sufficient time has passed for the node values to become more similar. Therefore, the rewiring fractures the network into fewer and larger clusters.

Furthermore, we have shown that heterogeneity changes the state the system reaches in equilibrium. Although the influence of heterogeneity is clearly visible, it is not as strong as we had anticipated. The heterogeneous case needs to be investigated further as we do not fully understand how heterogeneity in the rates influences the dynamics.

We have presented a definition for timescale separation in the case of homogeneous and therefore well defined rates, but we need an extended definition for the case of hetero-

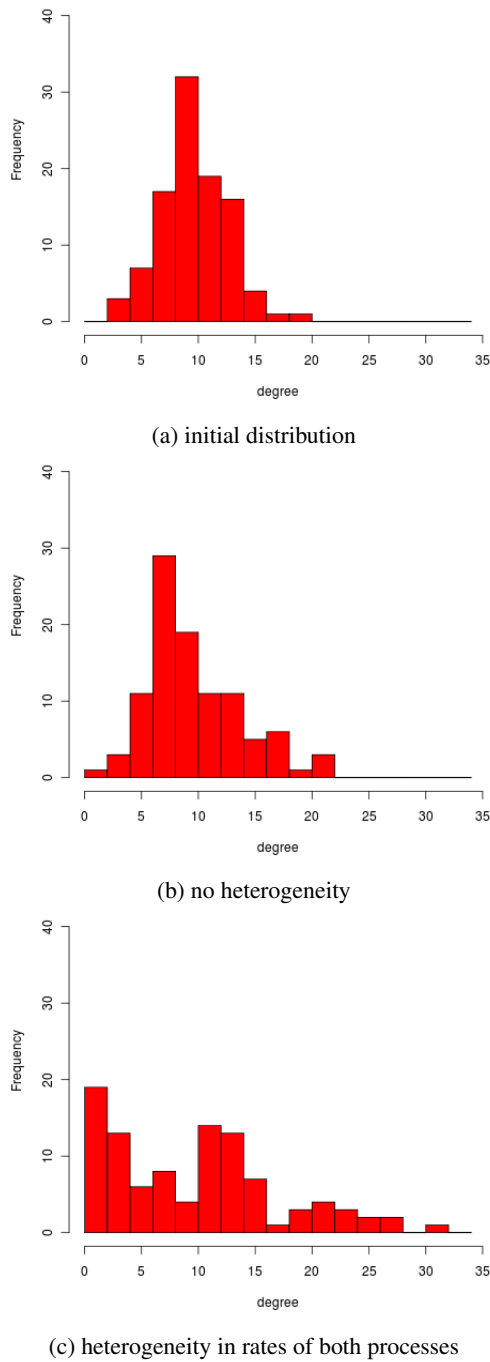


Figure 6: Histogram of the degree distribution under different conditions. The rate parameters used are  $V = 1$  and  $R = 2000$ . In the heterogeneous case the  $\alpha$  values are  $\alpha_V = 2.1$  and  $\alpha_R = -4.1$ .

geneous rates. There are of course further complications in real-world systems that we have not considered in the model presented here. For example, processes often have dynamic rates, i.e., the change of the rate is a process itself, perhaps influenced by the current state of the system.

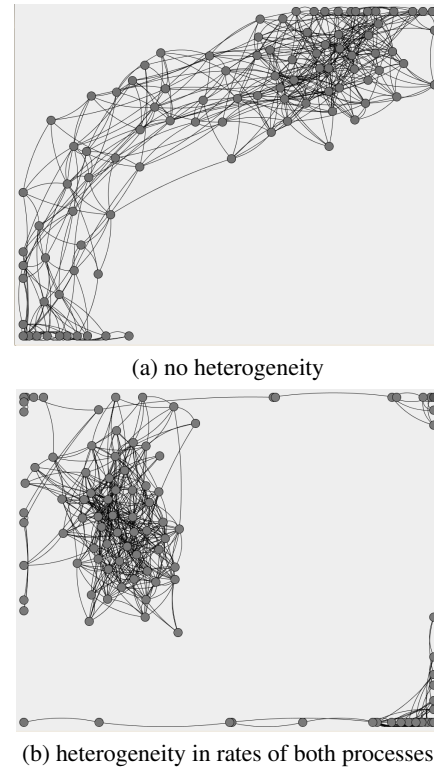


Figure 7: The effect of heterogeneity on network topology for  $V = 1$ ,  $R = 2000$ ,  $\alpha_V = 2.1$  and  $\alpha_R = -4.1$ .

## Conclusions

In this paper we have presented an initial investigation of timescale separation in adaptive networks, by identifying examples from the literature of different ways of dealing with multiple timescales and proposing a definition of timescale separation, based on the time taken by a system to reach equilibrium under the action of individual processes. Given this definition, we confirmed that, if the timescales of two processes are sufficiently separated, we can ignore their interaction. Where timescales do not separate cleanly, however, the system dynamics exhibit higher variability and hence become more difficult to predict. Heterogeneity complicates matters further as it can result in the system relaxing to different equilibria in comparison to the same system under homogeneous conditions. Where we can not be certain that the timescales are sufficiently separated in a system under consideration, we should expect the dynamics to be sensitive to the interplay between the timescales of the processes present.

## Acknowledgements

This work was supported by an EPSRC Doctoral Training Centre grant (EP/G03690X/1) and grant (EP/G036926/1.).



## References

- Ackley, D. H. and Littman, M. L. (1992). Interactions between learning and evolution. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II*, pages 487–509. Addison-Wesley, Redwood City, CA.
- Belew, R. K. and Mitchell, M., editors (1996). *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Addison-Wesley, Reading, MA.
- Blasius, B. and Gross, T. (2009). Dynamic and topological interplay in adaptive networks. In Schuster, H. G., editor, *Annual Reviews of Nonlinear Dynamics and Complexity*, volume 2, chapter 3, pages 63–106. Wiley-VCH, Weinheim.
- Bryden, J., Funk, S., Geard, N., Bullock, S., and Jansen, V. A. A. (2010). Stability in flux: community structure in dynamic networks. *Journal of the Royal Society Interface*, 7:1257–1274.
- Buckley, C., Bullock, S., and Cohen, N. (2004). Toward a dynamical systems analysis of neuromodulation. In Schaal, S., Ijspeert, A. J., Billard, A., Vijayakumar, S., Hallam, J., and Meyer, J.-A., editors, *Eighth International Conference on Simulation of Adaptive Behavior*, pages 334–343. MIT Press, Cambridge, MA.
- Buckley, C., Bullock, S., and Cohen, N. (2005). Timescale and stability in adaptive behaviour. In Bentley, P., Capcarrere, M., Freitas, A. A., Johnson, C. G., and Timmis, J., editors, *Eighth European Conference on Artificial Life*, pages 292–301. Springer, Berlin.
- Buckley, C. L. (2008). *A systemic analysis of the ideas immanent in neuromodulation*. PhD thesis, School of Electronics and Computer Science, University of Southampton, UK.
- Demirel, G., Prizak, R., Reddy, P. N., and Gross, T. (2011). Cyclic dominance in adaptive networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, pages 1–8.
- Funk, S. and Jansen, V. A. A. (2010). Interacting epidemics on overlay networks. *Physical Review E*, 81(3):036118.
- Gross, T. and Blasius, B. (2008). Adaptive coevolutionary networks: a review. *Journal of The Royal Society Interface*, 5(20):259–271.
- Gross, T. and Sayama, H., editors (2009). *Adaptive networks: Theory, Models and Applications*. Springer, New York.
- Harvey, I. and Bossomaier, T. (1997). Time out of joint: Attractors in asynchronous random Boolean networks. In Husbands, P. and Harvey, I., editors, *Proceedings of the Fourth European Conference on Artificial Life*, pages 67–75. MIT Press, Cambridge MA.
- Hinton, G. E. and Nowlan, S. J. (1987). How learning can guide evolution. *Complex Systems*, 1:495–502.
- Huberman, B. A. and Glance, N. S. (1993). Evolutionary games and computer simulations. *Proceedings of the National Academy of Sciences USA*, 90(16):7715–7718.
- Husbands, P., Philippides, A., Vargas, P., Buckley, C. L., Fine, P., Di Paolo, E., and O’Shea, M. (2010). Spatial, temporal, and modulatory factors affecting gasnet evolvability in a visually guided robotics task. *Complexity*, 16(2):35–44.
- Kauffman, S. A. (1993). *The Origins of Order*. Oxford University Press.
- Ke, H. and Yi, T. (2008). Information diffusion on adaptive network. *Chinese Physics B*, 17(10):3536.
- Kozma, B. and Barrat, A. (2008). Consensus formation on adaptive networks. *Physical Review E*, 77(1):016102.
- Newman, M. (2004). Power laws, Pareto distributions and Zipf’s law. *Arxiv preprint cond-mat/0412004*.
- Newman, M. E. J. (2002). The spread of epidemic disease on networks. *Physical Review E*, 66(1):016128.
- Nowak, M. A. and May, R. M. (1992). Evolutionary games and spatial chaos. *Nature*, 359(6398):826–829.
- Onnela, J. P. and Reed-Tsochas, F. (2009). The spontaneous emergence of social influence in online systems. *Proceedings of the National Academy of Sciences*, 107(43):18375–18380.
- Pacheco, J. M., Traulsen, A., and Nowak, M. A. (2006). Coevolution of strategy and structure in complex networks with dynamical linking. *Physical Review Letters*, 97(25):258103.
- Palla, G., Barabási, A., and Vicsek, T. (2007). Quantifying social group evolution. *Nature*, 446(7136):664–667.
- Powers, S. T., Penn, A. S., and Watson, R. A. (in press). The concurrent evolution of cooperation and the population structures that support it. *Evolution*.
- Sun, C. and Wang, S. (2008). Modeling adaptive behaviors on growing social networks. In *Fourth International Conference on Natural Computation*, volume 1, pages 465–469, Los Alamitos, CA, USA. IEEE Computer Society.
- Van Der Laan, J. and Hogeweg, P. (1995). Predator-prey coevolution: interactions across different timescales. *Proceedings: Biological Sciences*, 259(1354):35–42.
- Van Segbroeck, S., Santos, F. C., Lenaerts, T., and Pacheco, J. M. (2009). Reacting differently to adverse ties promotes cooperation in social networks. *Physical Review Letters*, 102(5):058105.
- Van Segbroeck, S., Santos, F. C., and Pacheco, J. M. (2010). Adaptive contact networks change effective disease infectiousness and dynamics. *PLoS Computational Biology*, 6(8):e1000895.
- Watson, R. A., Mills, R., and Buckley, C. L. (in press). Global adaptation in networks of selfish components: Emergent associative memory at the system scale. *Artificial Life*.
- Watts, D. J. (2002). A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences USA*, 99(9):5766–5771.
- Zhong, L.-X., Qiu, T., Ren, F., Li, P., and Chen, B. (2010). Time scales of epidemic spread and risk perception on adaptive networks. *Europhysics Letters*, 94(1):18004.
- Zhu, J.-F., Zhao, M., Yu, W., Zhou, C., and Wang, B.-H. (2010). Better synchronizability in generalized adaptive networks. *Physical Review E*, 81(2 Pt 2):026201.

Abbass, Hussein A.  
 Abe, Hirotake  
 Abgrall, Jean-François  
 Adamatzky, Andrew  
 Agostini, Davide  
 Antonioni, Alberto  
 Arranz, Jordi  
 Aston, Elizabeth  
 Aston, John  
 Balduzzi, David  
 Banzhaf, Wolfgang  
 Barandiaran, Xabier E.  
 Barata, José  
 Barnes, David J.  
 Bartlett, Stuart  
 Bedau, Mark A.  
 Belavkin, Roman V.  
 Bentley, Peter J.  
 Bernardet, Ulysses  
 Beslon, Guillaume  
 Bhalla, Navneet  
 Bigge, Bill  
 Bijak, Jakub  
 Birattari, Mauro  
 Blackwell, Tim  
 Borg, James M.  
 Bown, James  
 Boyle, Richard A.  
 Bredeche, Nicolas  
 Bremner, Paul  
 Bryson, David M.  
 Buchanan, Andrew J.  
 Buckley, Christopher L.  
 Buecheler, Thierry  
 Bullock, Seth  
 Cañamero, Lola  
 Carrara, Paolo  
 Caves, Leo S.  
 Chalmers, Devin W.  
 Channon, Alastair  
 Chevallier, Sylvain  
 Christensen, Anders Lyhne  
 Chu, Dominique  
 Clark, Ed  
 Clarke, Tim  
 Clune, Jeff  
 Coates, Richard  
 Cockburn, Denton  
 Collard, Philippe  
 Contreras, Diego A.  
 Correia, Luis  
 Costanza, Jole  
 Crailsheim, Karl  
 Cruse, Holk  
 Cussat-Blanc, Sylvain  
 Cutello, Vincenzo  
 Damiano, Luisa  
 Darabos, Christian  
 Day, Charles  
 De Lucrezia, Davide  
 de Souza, Tereza Perreira  
 Decraene, James  
 Dittrich, Peter  
 Doncaster, C. Patrick  
 Dorigo, Marco  
 Dragffy, Gabriel  
 Droop, Alastair  
 Ducommun, Bernard  
 Duff, Armin  
 Duthen, Yves  
 Dyer, Fred C.  
 Egbert, Matthew D.  
 Elena, Santiago F.  
 Enomoto, Koichiro

Erbas, Mehmet D.  
 Essam, Daryl L.  
 Farahani, Yasaman Majdabadi  
 Fatés, Nazim  
 Fernando, Chrisantha  
 Ferrante, Eliseo  
 Filisetti, Alessandro  
 Fisher, Jonathan M.  
 Francis, C. Cooper  
 Frei, Regina  
 Froese, Tom  
 Fuchsln, Rudolf M.  
 Fujii, Teruo  
 Furukawa, Masashi  
 Gautrais, Jacques  
 Gerlee, Philip  
 Gershenson, Carlos  
 Ghoneim, Amr S.  
 Gil, Debora  
 Giraldo, Jesus  
 Goldsby, Heather  
 Golestani, Abbas  
 Gordon-Smith, Chris  
 Grabowski, Laura M.  
 Gras, Robin  
 Graudenzi, Alex  
 Gudwin, Ricardo  
 Guest, Andrew  
 Gunji, Yukio  
 Gutiérrez, Claudio  
 Hamann, Heiko  
 Haruna, Taichi  
 Harvey, Inman  
 Hatcher, Mark  
 Hernandez, Valentina C.  
 Herrmann, J. Michael  
 Heys, Christopher  
 Hickinbotham, Simon  
 Hidalgo, Diana  
 Hiolle, Antoine  
 Honda, Manabu  
 Hong, Jewoo  
 Hoteit, Imad  
 Hourdakias, Emmanouil  
 Hoverd, Tim  
 Hubert, Julien  
 Husbands, Phil  
 Ibne, Sina Md  
 Iizuka, Hiroyuki  
 Iizuka, Kojiro  
 Ikegami, Takashi  
 Ishida, Takeshi  
 Jacob, Christian  
 Jaramillo, Sebastian  
 Jones, Daniel  
 Jost, Christian  
 Katada, Yoshiaki  
 Kawai, Norie  
 Kharma, Nawwaf  
 Khater, Marwa  
 Khor, Susan  
 Khuong, Anaïs  
 Kim, DaeEun  
 Kim, Joon  
 Knibbe, Carole  
 Knight, Christopher G.  
 Knox, Davis  
 Kobti, Ziad  
 Kolchinsky, Artemy  
 Krasnogor, Natalio  
 Kreyssig, Peter  
 Kuruma, Yutetsu  
 Kuyucu, Tüze  
 Kwon, Hyojung

Lancet, Doron  
 Le Gal, Christophe  
 Le Yaouanq, Sébastien  
 Lenski, Richard E.  
 Lenton, Tim M.  
 Letelier, Juan-Carlos  
 Li, Shuguang  
 Lima, Pedro U.  
 Lipson, Hod  
 Lizier, Joseph T.  
 Lobjois, Valérie  
 Lones, Michael A.  
 Lonigro, Rocky  
 Loula, Angelo  
 Luga, Hervé  
 Luisi, Pier Luigi  
 Lundh, Torbjörn  
 Maekawa, Tadao  
 Mariano, Pedro  
 Markovitch, Omer  
 Marriott, Chris  
 Martinoli, Alcherio  
 Martius, Georg  
 Mashayekhi, Morteza  
 Matsuda, Eiko  
 Matsui, Tetsuya  
 McCabe, Connor  
 McCrohon, Luke  
 McMullin, Barry  
 Mendonça, Ricardo  
 Mesmoudi, Salma  
 Miller, Jennifer M.  
 Mills, Rob  
 Misevic, Dusan  
 Miyake, Takuma  
 Mokon, Felicitas  
 Montagne, Kevin  
 Montanier, Jean-Marc  
 Moore, Jason H.  
 Moriguchi, Hirotaka  
 Moriyama, Toru  
 Murakami, Hisashi  
 Murdock, Jaimie  
 Nagle, Fintan  
 Nakamura, Keita  
 Nanas, Nikolaos  
 Nellis, Adam  
 Nguyen, Sarah  
 Nicosia, Giuseppe  
 Niizato, Takayuki  
 Nishina, Emi  
 Nishiyama, Yuta  
 Noble, Jason  
 Nunes, Luis  
 Ofria, Charles  
 Oka, Mizuki  
 Oliveira, Sancho  
 Oohashi, Tsutomu  
 Pacheco, Jorge M.  
 Packard, Norman H.  
 Padirac, Adrien  
 Pan, Qinxin  
 Parsons, David P.  
 Pascalie, Jonathan  
 Paugam-Moisy, Hélène  
 Pavone, Mario  
 Pay, Mungo  
 Payne, Joshua L.  
 Pennock, Robert T.  
 Pepper, Noah M.  
 Perdomo, Jonatan Gomez  
 Pereira, José N.  
 Pereira, Ulises  
 Perna, Andrea

Pestelacci, Enea	Virgo, Nathaniel
Pfeifer, Rolf	Vize, Peter D.
Philippides, Andrew	Vlassopoulos, Nikolaos
Pincioli, Carlo	Wang, X. Rosalind
Pipe, Tony	Watkins, Angela
Plasson, Raphaël	Watson, Richard A.
Polani, Daniel	Williams, Hywel T.P.
Poli, Irene	Williams, Lance R.
Portela, Arles Rodríguez	Wills, Peter R.
Powers, Simon T.	Winfield, Alan F.T.
Pritam, Siddharth	Witkowski, Olaf
Prokopenko, Mikhail	Wong, Chun
Queiroz, João	Wu, Shelly Xiaonan
Redou, Pascal	Yaeger, Larry S.
Reynaert, Bryan	Yamamoto, Masahito
Rieffel, John	Yoneda, Keisuke
Rocha, Luis M.	Yoshida, Masasuke
Roche, David	Yosinski, Jason
Rojas-Camaggi, David	Young, Peter
Rondelez, Yannick	Yu, Seung-Eun
Rossi, Louis F.	Yu, Tina
Salehi, Elham	Yuan, Jianping
Salge, Christoph	Yun, Juyeol
Samie, Mohammad	Zagal, Juan Cristóbal
Sanchez-Fibla, Marti	Zammataro, Luca
Sanjuan, Rafael	Zu Erbach-Schoenberg, Elisabeth
Santana, Pedro	
Santos, Francisco C.	
Sapeluk, Andrew	
Sato, Yuki	
Sayama, Hiroki	
Schilling, Malte	
Schmickl, Thomas	
Schoenauer, Marc	
Schramm, Lisa	
Scott, Ryan	
Sebag, Michèle	
Sedehi, Meisam Hosseini	
Sendhoff, Bernhard	
Serra, Roberto	
Shimohara, Katsunori	
Silva, Porfirio	
Silverman, Eric	
Sirmai, Jean	
Skyrms, Brian	
Soto-Andrade, Jorge	
Souza, Max O.	
Spranger, Michael	
Stano, Pasquale	
Stepney, Susan	
Stranieri, Alessandro	
Suzuki, Ikuo	
Suzuki, Keisuke	
Suzuki, Toshiharu	
Tanev, Ivan	
Theraulaz, Guy	
Thorniley, James	
Tisseau, Jacques	
Toda, Masashi	
Tomassini, Marco	
Tominaga, Kazuto	
Tomko, Nicholas	
Trahanias, Panos	
Trianni, Vito	
Turgut, Ali E.	
Tyler, Anna L.	
Tyrrell, Andy M.	
Ueda, Takuya	
Ueno, Osamu	
Uno, Ryoko	
Varin, Luc	
Veloz, Tomas	
Verschure, Paul F.M.J.	
Villani, Marco	